Información de tablas en supabase:

1. Tablas y Columnas


```
[
  {
    "table_name": "line_sessions",
    "column_name": "session_id",
    "data_type": "character varying",
    "udt_name": "varchar",
    "character_maximum_length": 100,
    "is_nullable": "NO",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "line_sessions",
    "column_name": "restaurant_id",
    "data_type": "integer",
    "udt_name": "int4",
    "character_maximum_length": null,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "line_sessions",
    "column_name": "person_id",
    "data_type": "integer",
    "udt_name": "int4",
    "character_maximum_length": null,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "line_sessions",
    "column_name": "parent_session_id",
    "data_type": "character varying",
    "udt_name": "varchar",
    "character_maximum_length": 100,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
```

```
  "table_name": "line_sessions",
  "column_name": "channel_type",
  "data_type": "USER-DEFINED",
  "udt_name": "channel_type_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "channel_id",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 100,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "platform_session_id",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 255,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_start",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_end",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
```

```
},
{
  "table_name": "line_sessions",
  "column_name": "duration_minutes",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "last_activity_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "message_count",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "user_messages",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "bot_messages",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
```

```
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "response_time_avg",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_complexity",
  "data_type": "USER-DEFINED",
  "udt_name": "session_complexity_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "inquiry_type",
  "data_type": "USER-DEFINED",
  "udt_name": "inquiry_type_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "primary_intent",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 100,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "products_discussed",
  "data_type": "jsonb",
  "udt_name": "jsonb",
```

```
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "suppliers_mentioned",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "categories_explored",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "ai_model_version",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 50,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "total_ai_calls",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "embeddings_used",
```

```
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "recommendations_made",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "recommendations_accepted",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "dominant_language",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 10,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "sentiment_overall",
  "data_type": "USER-DEFINED",
  "udt_name": "sentiment_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
```

```
  "table_name": "line_sessions",
  "column_name": "sentiment_scores",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "emotion_detected",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "urgency_level",
  "data_type": "USER-DEFINED",
  "udt_name": "urgency_level_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "preferences_captured",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "new_preferences",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
```

```
},
{
  "table_name": "line_sessions",
  "column_name": "preference_confidence",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "behavioral_signals",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "learning_opportunities",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_goal_achieved",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "conversion_occurred",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
```

```
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "order_value",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "satisfaction_level",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "resolution_status",
  "data_type": "USER-DEFINED",
  "udt_name": "resolution_status_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "follow_up_required",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "follow_up_type",
  "data_type": "USER-DEFINED",
  "udt_name": "follow_up_type_enum",
```

```
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "follow_up_scheduled_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "follow_up_completed",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_type",
  "data_type": "USER-DEFINED",
  "udt_name": "session_type_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "learning_value",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "pattern_match",
```

```
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "anomalies_detected",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "improvement_suggestions",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "escalation_required",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "escalation_reason",
  "data_type": "text",
  "udt_name": "text",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
```

```
  "table_name": "line_sessions",
  "column_name": "escalated_to",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 100,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "escalation_resolved",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "created_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "updated_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "session_notes",
  "data_type": "text",
  "udt_name": "text",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
```

```
},
{
  "table_name": "line_sessions",
  "column_name": "internal_tags",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "quality_score",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "awaiting_continuation",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "line_sessions",
  "column_name": "continuation_timestamp",
  "data_type": "timestamp without time zone",
  "udt_name": "timestamp",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "id",
  "data_type": "bigint",
  "udt_name": "int8",
  "character_maximum_length": null,
  "is_nullable": "NO",
```

```
  "column_default": "nextval('master_list_id_seq'::regclass)",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "product_name",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 255,
  "is_nullable": "NO",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "product_description",
  "data_type": "text",
  "udt_name": "text",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "brand",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 255,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "barcode_ean",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 50,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "base_category_id",
  "data_type": "integer",
  "udt_name": "int4",
```

```
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "product_category_id",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "product_tags",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "embedding_vector",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "embedding_version",
  "data_type": "character varying",
  "udt_name": "varchar",
  "character_maximum_length": 50,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "embedding_created_at",
```

```
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "search_frequency",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "match_confidence_avg",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "alternative_names",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "synonym_variations",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
```

```
  "table_name": "master_list",
  "column_name": "common_misspellings",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "quality_tier",
  "data_type": "USER-DEFINED",
  "udt_name": "quality_tier_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "specifications",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "seasonal_availability",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "peak_season_months",
  "data_type": "jsonb",
  "udt_name": "jsonb",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
```

```
},
{
  "table_name": "master_list",
  "column_name": "availability_score",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "total_orders",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "last_ordered_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "popularity_score",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "0",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "trend_indicator",
  "data_type": "USER-DEFINED",
  "udt_name": "trend_indicator_enum",
  "character_maximum_length": null,
  "is_nullable": "YES",
```

```
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "created_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "updated_at",
  "data_type": "timestamp with time zone",
  "udt_name": "timestamptz",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "now()",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "created_by",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "is_active",
  "data_type": "boolean",
  "udt_name": "bool",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "true",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "is_verified",
  "data_type": "boolean",
  "udt_name": "bool",
```

```
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": "false",
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "embedding_vector_v2",
  "data_type": "USER-DEFINED",
  "udt_name": "vector",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "restaurant_id",
  "data_type": "integer",
  "udt_name": "int4",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": null
},
{
  "table_name": "master_list",
  "column_name": "wallet_spend_share_position",
  "data_type": "numeric",
  "udt_name": "numeric",
  "character_maximum_length": null,
  "is_nullable": "YES",
  "column_default": null,
  "column_comment": "wallet_spend_share_position"
},
{
  "table_name": "pricing_history",
  "column_name": "id",
  "data_type": "bigint",
  "udt_name": "int8",
  "character_maximum_length": null,
  "is_nullable": "NO",
  "column_default": "nextval('pricing_history_id_seq'::regclass)",
  "column_comment": null
},
{
  "table_name": "pricing_history",
  "column_name": "supplier_id",
```

```
    "data_type": "integer",
    "udt_name": "int4",
    "character_maximum_length": null,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "pricing_history",
    "column_name": "master_list_id",
    "data_type": "bigint",
    "udt_name": "int8",
    "character_maximum_length": null,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "pricing_history",
    "column_name": "supplier_mapped_product_id",
    "data_type": "bigint",
    "udt_name": "int8",
    "character_maximum_length": null,
    "is_nullable": "YES",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "pricing_history",
    "column_name": "unit_price",
    "data_type": "numeric",
    "udt_name": "numeric",
    "character_maximum_length": null,
    "is_nullable": "NO",
    "column_default": null,
    "column_comment": null
  },
  {
    "table_name": "pricing_history",
    "column_name": "currency",
    "data_type": "character varying",
    "udt_name": "varchar",
    "character_maximum_length": 3,
    "is_nullable": "YES",
    "column_default": "'EUR'::character varying",
    "column_comment": null
  }
]
```

2. ENUMs con sus Valores


```
[
  {
    "enum_name": "ai_comfort_level_enum",
    "enum_values": "{low,medium,high}"
  },
  {
    "enum_name": "availability_status_enum",
    "enum_values": "{available,limited,out_of_stock,discontinued}"
  },
  {
    "enum_name": "brand_vs_master_enum",
    "enum_values": "{same,different,generic}"
  },
  {
    "enum_name": "business_type_enum",
    "enum_values": "{wholesaler,distributor,manufacturer,local_producer}"
  },
  {
    "enum_name": "category_maturity_enum",
    "enum_values": "{emerging,growing,mature,declining}"
  },
  {
    "enum_name": "channel_type_enum",
    "enum_values": "{whatsapp,telegram,web,mobile,phone,email}"
  },
  {
    "enum_name": "communication_style_enum",
    "enum_values": "{formal,casual,technical}"
  },
  {
    "enum_name": "company_size_enum",
    "enum_values": "{small,medium,large}"
  },
  {
    "enum_name": "compliance_status_enum",
    "enum_values": "{compliant,warning,failed}"
  },
  {
    "enum_name": "contact_method_enum",
    "enum_values": "{email,whatsapp,phone,website,portal,fax,in_person}"
  },
  {
    "enum_name": "credit_rating_enum",
    "enum_values": "{excellent,good,fair,poor}"
```

```
},
{
  "enum_name": "customer_base_size_enum",
  "enum_values": "{small,medium,large}"
},
{
  "enum_name": "customer_segment_enum",
  "enum_values": "{premium,standard,budget,new,loyal,at_risk}"
},
{
  "enum_name": "data_source_enum",
  "enum_values": "{supplier,scraping,manual,api,import}"
},
{
  "enum_name": "decision_speed_enum",
  "enum_values": "{fast,normal,slow}"
},
{
  "enum_name": "delivery_status_enum",
  "enum_values": "{scheduled,in_transit,delivered,failed}"
},
{
  "enum_name": "demand_level_enum",
  "enum_values": "{low,normal,high,peak}"
},
{
  "enum_name": "engagement_level_enum",
  "enum_values": "{low,medium,high}"
},
{
  "enum_name": "feedback_frequency_enum",
  "enum_values": "{never,rare,occasional,frequent}"
},
{
  "enum_name": "follow_up_type_enum",
  "enum_values": "{call,email,whatsapp,none}"
},
{
  "enum_name": "fulfillment_status_enum",
  "enum_values": "{full,partial,failed}"
},
{
  "enum_name": "growth_trajectory_enum",
  "enum_values": "{growing,stable,declining}"
},
{
  "enum_name": "inquiry_type_enum",
```

```
    "enum_values":
"{product_search,price_check,order_placement,support,information,complaint}"
  },
  {
    "enum_name": "learning_speed_enum",
    "enum_values": "{slow,normal,fast}"
  },
  {
    "enum_name": "lifecycle_stage_enum",
    "enum_values": "{prospect,new,growing,mature,declining}"
  },
  {
    "enum_name": "mapping_method_enum",
    "enum_values": "{manual,ai_matched,verified,customer_confirmed}"
  },
  {
    "enum_name": "market_position_enum",
    "enum_values": "{premium,mid-market,budget}"
  },
  {
    "enum_name": "market_position_price_enum",
    "enum_values": "{below,at,above,premium}"
  },
  {
    "enum_name": "negotiation_style_enum",
    "enum_values": "{aggressive,balanced,passive}"
  },
  {
    "enum_name": "order_complexity_enum",
    "enum_values": "{simple,moderate,complex}"
  },
  {
    "enum_name": "order_status_enum",
    "enum_values": "{pending,confirmed,prepared,shipped,delivered,cancelled}"
  },
  {
    "enum_name": "order_type_enum",
    "enum_values": "{regular,emergency,bulk}"
  },
  {
    "enum_name": "payment_status_enum",
    "enum_values": "{pending,paid,overdue}"
  },
  {
    "enum_name": "price_competitiveness_enum",
    "enum_values": "{low,competitive,high}"
  },
  {
```

```
  "enum_name": "price_trend_enum",
  "enum_values": "{up,down,stable}"
},
{
  "enum_name": "pricing_update_frequency_enum",
  "enum_values": "{daily,weekly,monthly,quarterly,seasonal,irregular}"
},
{
  "enum_name": "priority_level_enum",
  "enum_values": "{low,normal,high,urgent}"
},
{
  "enum_name": "quality_tier_enum",
  "enum_values": "{premium,standard,economy}"
},
{
  "enum_name": "quality_vs_master_enum",
  "enum_values": "{same,better,worse,unknown}"
},
{
  "enum_name": "research_depth_enum",
  "enum_values": "{minimal,moderate,thorough}"
},
{
  "enum_name": "resolution_status_enum",
  "enum_values": "{resolved,pending,escalated}"
},
{
  "enum_name": "restaurant_type_enum",
  "enum_values": "{restaurant,cafe,bistro,brasserie,hotel,catering}"
},
{
  "enum_name": "revenue_range_enum",
  "enum_values": "{0-100k,100k-500k,500k+}"
},
{
  "enum_name": "risk_level_enum",
  "enum_values": "{low,medium,high}"
},
{
  "enum_name": "role_category_enum",
  "enum_values": "{owner,manager,chef,sous_chef,buyer,staff}"
},
{
  "enum_name": "seniority_level_enum",
  "enum_values": "{junior,mid,senior,exec}"
},
{
```

```
  "enum_name": "sentiment_enum",
  "enum_values": "{positive,neutral,negative}"
 },
 {
  "enum_name": "session_complexity_enum",
  "enum_values": "{simple,moderate,complex}"
 },
 {
  "enum_name": "session_type_enum",
  "enum_values": "{discovery,negotiation,transactional,support,setup,purchase}"
 },
 {
  "enum_name": "size_vs_master_enum",
  "enum_values": "{same,larger,smaller}"
 },
 {
  "enum_name": "supply_level_enum",
  "enum_values": "{shortage,low,normal,surplus}"
 },
 {
  "enum_name": "trend_indicator_enum",
  "enum_values": "{rising,stable,declining}"
 },
 {
  "enum_name": "urgency_level_enum",
  "enum_values": "{low,normal,high,critical}"
 },
 {
  "enum_name": "user_type_enum",
  "enum_values": "{power_user,regular,casual}"
 },
 {
  "enum_name": "value_tier_enum",
  "enum_values": "{high,medium,low}"
 },
 {
  "enum_name": "verification_status_enum",
  "enum_values": "{pending,verified,rejected}"
 },
 {
  "enum_name": "verification_status_price_enum",
  "enum_values": "{unverified,verified,disputed}"
 }
]
```


3. Primary Keys

```
[
  {
    "table_name": "line_sessions",
    "column_name": "session_id",
    "constraint_name": "line_sessions_pkey"
  },
  {
    "table_name": "master_list",
    "column_name": "id",
    "constraint_name": "master_list_pkey"
  },
  {
    "table_name": "pricing_history",
    "column_name": "id",
    "constraint_name": "pricing_history_pkey"
  },
  {
    "table_name": "product_categories",
    "column_name": "id",
    "constraint_name": "product_categories_pkey"
  },
  {
    "table_name": "purchase_orders",
    "column_name": "order_id",
    "constraint_name": "purchase_orders_pkey"
  },
  {
    "table_name": "restaurant_people",
    "column_name": "id",
    "constraint_name": "restaurant_people_pkey"
  },
  {
    "table_name": "restaurant_product_preferences",
    "column_name": "id",
    "constraint_name": "restaurant_product_preferences_pkey"
  },
  {
    "table_name": "restaurant_product_preferences_history",
    "column_name": "id",
    "constraint_name": "restaurant_product_preferences_history_pkey"
  },
  {
    "table_name": "restaurants",
    "column_name": "id",
    "constraint_name": "restaurants_pkey"
  },
  {
    "table_name": "spatial_ref_sys",
```

```
    "column_name": "srid",
    "constraint_name": "spatial_ref_sys_pkey"
  },
  {
    "table_name": "supplier_mapped_products",
    "column_name": "id",
    "constraint_name": "supplier_mapped_products_pkey"
  },
  {
    "table_name": "suppliers",
    "column_name": "id",
    "constraint_name": "suppliers_pkey"
  },
  {
    "table_name": "user_preferences",
    "column_name": "id",
    "constraint_name": "user_preferences_pkey"
  }
]
```


4. Foreign Keys y Relaciones


```
[
  {
    "tabla_origen": "line_sessions",
    "columna_origen": "parent_session_id",
    "tabla_destino": "line_sessions",
    "columna_destino": "session_id",
    "constraint_name": "line_sessions_parent_session_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  },
  {
    "tabla_origen": "line_sessions",
    "columna_origen": "person_id",
    "tabla_destino": "restaurant_people",
    "columna_destino": "id",
    "constraint_name": "line_sessions_person_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  },
  {
    "tabla_origen": "line_sessions",
    "columna_origen": "restaurant_id",
    "tabla_destino": "restaurants",
    "columna_destino": "id",
    "constraint_name": "line_sessions_restaurant_id_fkey",
```

```
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "master_list",
  "columna_origen": "product_category_id",
  "tabla_destino": "product_categories",
  "columna_destino": "id",
  "constraint_name": "master_list_product_category_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "master_list",
  "columna_origen": "restaurant_id",
  "tabla_destino": "restaurants",
  "columna_destino": "id",
  "constraint_name": "master_list_restaurant_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "pricing_history",
  "columna_origen": "master_list_id",
  "tabla_destino": "master_list",
  "columna_destino": "id",
  "constraint_name": "pricing_history_master_list_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "pricing_history",
  "columna_origen": "supplier_id",
  "tabla_destino": "suppliers",
  "columna_destino": "id",
  "constraint_name": "pricing_history_supplier_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "pricing_history",
  "columna_origen": "supplier_mapped_product_id",
  "tabla_destino": "supplier_mapped_products",
  "columna_destino": "id",
  "constraint_name": "pricing_history_supplier_mapped_product_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
```

```
{
  "tabla_origen": "product_categories",
  "columna_origen": "parent_category_id",
  "tabla_destino": "product_categories",
  "columna_destino": "id",
  "constraint_name": "product_categories_parent_category_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "purchase_orders",
  "columna_origen": "ordered_by_person_id",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "purchase_orders_ordered_by_person_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "purchase_orders",
  "columna_origen": "restaurant_id",
  "tabla_destino": "restaurants",
  "columna_destino": "id",
  "constraint_name": "purchase_orders_restaurant_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "purchase_orders",
  "columna_origen": "session_id",
  "tabla_destino": "line_sessions",
  "columna_destino": "session_id",
  "constraint_name": "purchase_orders_session_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "purchase_orders",
  "columna_origen": "supplier_id",
  "tabla_destino": "suppliers",
  "columna_destino": "id",
  "constraint_name": "purchase_orders_supplier_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_people",
  "columna_origen": "reports_to_id",
```

```
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_people_reports_to_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_people",
  "columna_origen": "restaurant_id",
  "tabla_destino": "restaurants",
  "columna_destino": "id",
  "constraint_name": "restaurant_people_restaurant_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "brand_preferences_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_brand_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "frequency_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_frequency_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "master_list_id",
  "tabla_destino": "master_list",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_master_list_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "payment_preference_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_payment_added_by_fkey",
```

```
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "price_preference_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_price_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "quality_preference_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_quality_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "quantity_preference_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_quantity_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "restaurant_id",
  "tabla_destino": "restaurants",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_restaurant_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "specification_preference_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_spec_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
```

```
{
  "tabla_origen": "restaurant_product_preferences",
  "columna_origen": "tracked_only_added_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_tracked_added_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences_history",
  "columna_origen": "changed_by",
  "tabla_destino": "restaurant_people",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_history_changed_by_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "restaurant_product_preferences_history",
  "columna_origen": "preference_id",
  "tabla_destino": "restaurant_product_preferences",
  "columna_destino": "id",
  "constraint_name": "restaurant_product_preferences_history_preference_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "CASCADE"
},
{
  "tabla_origen": "restaurant_product_preferences_history",
  "columna_origen": "session_id",
  "tabla_destino": "line_sessions",
  "columna_destino": "session_id",
  "constraint_name": "restaurant_product_preferences_history_session_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "supplier_mapped_products",
  "columna_origen": "master_list_id",
  "tabla_destino": "master_list",
  "columna_destino": "id",
  "constraint_name": "supplier_mapped_products_master_list_id_fkey",
  "update_rule": "NO ACTION",
  "delete_rule": "NO ACTION"
},
{
  "tabla_origen": "supplier_mapped_products",
  "columna_origen": "supplier_id",
```

```
    "tabla_destino": "suppliers",
    "columna_destino": "id",
    "constraint_name": "supplier_mapped_products_supplier_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  },
  {
    "tabla_origen": "user_preferences",
    "columna_origen": "master_list_id",
    "tabla_destino": "master_list",
    "columna_destino": "id",
    "constraint_name": "user_preferences_master_list_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  },
  {
    "tabla_origen": "user_preferences",
    "columna_origen": "person_id",
    "tabla_destino": "restaurant_people",
    "columna_destino": "id",
    "constraint_name": "user_preferences_person_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  },
  {
    "tabla_origen": "user_preferences",
    "columna_origen": "restaurant_id",
    "tabla_destino": "restaurants",
    "columna_destino": "id",
    "constraint_name": "user_preferences_restaurant_id_fkey",
    "update_rule": "NO ACTION",
    "delete_rule": "NO ACTION"
  }
]
```


5. Índices


```
[
  {
    "schemaname": "public",
    "tablename": "line_sessions",
    "indexname": "idx_line_sessions_date",
    "indexdef": "CREATE INDEX idx_line_sessions_date ON public.line_sessions USING
btree (session_start DESC)"
  },
  {
    "schemaname": "public",
    "tablename": "line_sessions",
```

```
    "indexname": "idx_line_sessions_person",
    "indexdef": "CREATE INDEX idx_line_sessions_person ON public.line_sessions USING
btree (person_id)"
  },
  {
    "schemaname": "public",
    "tablename": "line_sessions",
    "indexname": "idx_line_sessions_restaurant",
    "indexdef": "CREATE INDEX idx_line_sessions_restaurant ON public.line_sessions
USING btree (restaurant_id)"
  },
  {
    "schemaname": "public",
    "tablename": "line_sessions",
    "indexname": "line_sessions_pkey",
    "indexdef": "CREATE UNIQUE INDEX line_sessions_pkey ON public.line_sessions
USING btree (session_id)"
  },
  {
    "schemaname": "public",
    "tablename": "master_list",
    "indexname": "idx_master_list_active",
    "indexdef": "CREATE INDEX idx_master_list_active ON public.master_list USING btree
(is_active) WHERE (is_active = true)"
  },
  {
    "schemaname": "public",
    "tablename": "master_list",
    "indexname": "idx_master_list_category",
    "indexdef": "CREATE INDEX idx_master_list_category ON public.master_list USING btree
(product_category_id)"
  },
  {
    "schemaname": "public",
    "tablename": "master_list",
    "indexname": "idx_master_list_embedding_v2",
    "indexdef": "CREATE INDEX idx_master_list_embedding_v2 ON public.master_list
USING hnsw (embedding_vector_v2 vector_cosine_ops)"
  },
  {
    "schemaname": "public",
    "tablename": "master_list",
    "indexname": "idx_master_list_search_freq",
    "indexdef": "CREATE INDEX idx_master_list_search_freq ON public.master_list USING
btree (search_frequency DESC)"
  },
  {
    "schemaname": "public",
```

```
    "tablename": "master_list",
    "indexname": "master_list_pkey",
    "indexdef": "CREATE UNIQUE INDEX master_list_pkey ON public.master_list USING
btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "pricing_history",
    "indexname": "idx_pricing_history_date",
    "indexdef": "CREATE INDEX idx_pricing_history_date ON public.pricing_history USING
btree (effective_date DESC)"
  },
  {
    "schemaname": "public",
    "tablename": "pricing_history",
    "indexname": "idx_pricing_history_product",
    "indexdef": "CREATE INDEX idx_pricing_history_product ON public.pricing_history
USING btree (master_list_id)"
  },
  {
    "schemaname": "public",
    "tablename": "pricing_history",
    "indexname": "idx_pricing_history_supplier",
    "indexdef": "CREATE INDEX idx_pricing_history_supplier ON public.pricing_history
USING btree (supplier_id)"
  },
  {
    "schemaname": "public",
    "tablename": "pricing_history",
    "indexname": "pricing_history_pkey",
    "indexdef": "CREATE UNIQUE INDEX pricing_history_pkey ON public.pricing_history
USING btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "product_categories",
    "indexname": "idx_product_categories_active",
    "indexdef": "CREATE INDEX idx_product_categories_active ON
public.product_categories USING btree (is_active) WHERE (is_active = true)"
  },
  {
    "schemaname": "public",
    "tablename": "product_categories",
    "indexname": "idx_product_categories_parent",
    "indexdef": "CREATE INDEX idx_product_categories_parent ON
public.product_categories USING btree (parent_category_id)"
  },
  {
```

```
    "schemaname": "public",
    "tablename": "product_categories",
    "indexname": "product_categories_category_slug_key",
    "indexdef": "CREATE UNIQUE INDEX product_categories_category_slug_key ON
public.product_categories USING btree (category_slug)"
  },
  {
    "schemaname": "public",
    "tablename": "product_categories",
    "indexname": "product_categories_pkey",
    "indexdef": "CREATE UNIQUE INDEX product_categories_pkey ON
public.product_categories USING btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "purchase_orders",
    "indexname": "idx_purchase_orders_date",
    "indexdef": "CREATE INDEX idx_purchase_orders_date ON public.purchase_orders
USING btree (order_date DESC)"
  },
  {
    "schemaname": "public",
    "tablename": "purchase_orders",
    "indexname": "idx_purchase_orders_restaurant",
    "indexdef": "CREATE INDEX idx_purchase_orders_restaurant ON public.purchase_orders
USING btree (restaurant_id)"
  },
  {
    "schemaname": "public",
    "tablename": "purchase_orders",
    "indexname": "idx_purchase_orders_supplier",
    "indexdef": "CREATE INDEX idx_purchase_orders_supplier ON public.purchase_orders
USING btree (supplier_id)"
  },
  {
    "schemaname": "public",
    "tablename": "purchase_orders",
    "indexname": "purchase_orders_pkey",
    "indexdef": "CREATE UNIQUE INDEX purchase_orders_pkey ON public.purchase_orders
USING btree (order_id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_people",
    "indexname": "idx_restaurant_people_active",
    "indexdef": "CREATE INDEX idx_restaurant_people_active ON public.restaurant_people
USING btree (is_active) WHERE (is_active = true)"
  },
```

```
 {
   "schemaname": "public",
   "tablename": "restaurant_people",
   "indexname": "idx_restaurant_people_restaurant",
   "indexdef": "CREATE INDEX idx_restaurant_people_restaurant ON
public.restaurant_people USING btree (restaurant_id)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_people",
   "indexname": "restaurant_people_pkey",
   "indexdef": "CREATE UNIQUE INDEX restaurant_people_pkey ON
public.restaurant_people USING btree (id)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_product_preferences",
   "indexname": "idx_restaurant_product_preferences_active",
   "indexdef": "CREATE INDEX idx_restaurant_product_preferences_active ON
public.restaurant_product_preferences USING btree (is_active) WHERE (is_active = true)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_product_preferences",
   "indexname": "idx_restaurant_product_preferences_product",
   "indexdef": "CREATE INDEX idx_restaurant_product_preferences_product ON
public.restaurant_product_preferences USING btree (master_list_id)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_product_preferences",
   "indexname": "idx_restaurant_product_preferences_restaurant",
   "indexdef": "CREATE INDEX idx_restaurant_product_preferences_restaurant ON
public.restaurant_product_preferences USING btree (restaurant_id)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_product_preferences",
   "indexname": "restaurant_product_preferences_pkey",
   "indexdef": "CREATE UNIQUE INDEX restaurant_product_preferences_pkey ON
public.restaurant_product_preferences USING btree (id)"
 },
 {
   "schemaname": "public",
   "tablename": "restaurant_product_preferences",
   "indexname": "restaurant_product_preferences_unique",
   "indexdef": "CREATE UNIQUE INDEX restaurant_product_preferences_unique ON
public.restaurant_product_preferences USING btree (restaurant_id, master_list_id)"
```

```
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_changed_at",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_changed_at
ON public.restaurant_product_preferences_history USING btree (changed_at DESC)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_changed_by",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_changed_by
ON public.restaurant_product_preferences_history USING btree (changed_by)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_preference",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_preference ON
public.restaurant_product_preferences_history USING btree (preference_id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_product",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_product ON
public.restaurant_product_preferences_history USING btree (master_list_id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_restaurant",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_restaurant ON
public.restaurant_product_preferences_history USING btree (restaurant_id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "idx_restaurant_product_preferences_history_session",
    "indexdef": "CREATE INDEX idx_restaurant_product_preferences_history_session ON
public.restaurant_product_preferences_history USING btree (session_id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurant_product_preferences_history",
    "indexname": "restaurant_product_preferences_history_pkey",
```

```
    "indexdef": "CREATE UNIQUE INDEX restaurant_product_preferences_history_pkey ON public.restaurant_product_preferences_history USING btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurants",
    "indexname": "idx_restaurants_active",
    "indexdef": "CREATE INDEX idx_restaurants_active ON public.restaurants USING btree (is_active) WHERE (is_active = true)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurants",
    "indexname": "idx_restaurants_location",
    "indexdef": "CREATE INDEX idx_restaurants_location ON public.restaurants USING gist (coordinates) WHERE (coordinates IS NOT NULL)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurants",
    "indexname": "idx_restaurants_segment",
    "indexdef": "CREATE INDEX idx_restaurants_segment ON public.restaurants USING btree (customer_segment)"
  },
  {
    "schemaname": "public",
    "tablename": "restaurants",
    "indexname": "restaurants_pkey",
    "indexdef": "CREATE UNIQUE INDEX restaurants_pkey ON public.restaurants USING btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "spatial_ref_sys",
    "indexname": "spatial_ref_sys_pkey",
    "indexdef": "CREATE UNIQUE INDEX spatial_ref_sys_pkey ON public.spatial_ref_sys USING btree (srid)"
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
    "indexname": "idx_supplier_mapped_products_confidence",
    "indexdef": "CREATE INDEX idx_supplier_mapped_products_confidence ON public.supplier_mapped_products USING btree (mapping_confidence DESC)"
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
```

```
    "indexname": "idx_supplier_mapped_products_master",
    "indexdef": "CREATE INDEX idx_supplier_mapped_products_master ON
public.supplier_mapped_products USING btree (master_list_id)"
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
    "indexname": "idx_supplier_mapped_products_price",
    "indexdef": "CREATE INDEX idx_supplier_mapped_products_price ON
public.supplier_mapped_products USING btree (current_unit_price)"
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
    "indexname": "idx_supplier_mapped_products_supplier",
    "indexdef": "CREATE INDEX idx_supplier_mapped_products_supplier ON
public.supplier_mapped_products USING btree (supplier_id)"
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
    "indexname": "supplier_mapped_products_pkey",
    "indexdef": "CREATE UNIQUE INDEX supplier_mapped_products_pkey ON
public.supplier_mapped_products USING btree (id)"
  },
  {
    "schemaname": "public",
    "tablename": "suppliers",
    "indexname": "idx_suppliers_active",
    "indexdef": "CREATE INDEX idx_suppliers_active ON public.suppliers USING btree
(is_active) WHERE (is_active = true)"
  },
  {
    "schemaname": "public",
    "tablename": "suppliers",
    "indexname": "idx_suppliers_contact_method",
    "indexdef": "CREATE INDEX idx_suppliers_contact_method ON public.suppliers USING
btree (preferred_contact_method)"
  },
  {
    "schemaname": "public",
    "tablename": "suppliers",
    "indexname": "idx_suppliers_location",
    "indexdef": "CREATE INDEX idx_suppliers_location ON public.suppliers USING gist
(coordinates) WHERE (coordinates IS NOT NULL)"
  },
  {
    "schemaname": "public",
```

```
    "tablename": "suppliers",
    "indexname": "suppliers_pkey",
    "indexdef": "CREATE UNIQUE INDEX suppliers_pkey ON public.suppliers USING btree
(id)"
  },
  {
    "schemaname": "public",
    "tablename": "user_preferences",
    "indexname": "idx_user_preferences_person",
    "indexdef": "CREATE INDEX idx_user_preferences_person ON public.user_preferences
USING btree (person_id)"
  },
  {
    "schemaname": "public",
    "tablename": "user_preferences",
    "indexname": "idx_user_preferences_restaurant",
    "indexdef": "CREATE INDEX idx_user_preferences_restaurant ON
public.user_preferences USING btree (restaurant_id)"
  },
  {
    "schemaname": "public",
    "tablename": "user_preferences",
    "indexname": "idx_user_preferences_type",
    "indexdef": "CREATE INDEX idx_user_preferences_type ON public.user_preferences
USING btree (preference_type)"
  },
  {
    "schemaname": "public",
    "tablename": "user_preferences",
    "indexname": "user_preferences_pkey",
    "indexdef": "CREATE UNIQUE INDEX user_preferences_pkey ON
public.user_preferences USING btree (id)"
  }
]
```


6. Triggers


```
[
  {
    "trigger_schema": "public",
    "trigger_name": "update_line_sessions_updated_at",
    "event_manipulation": "UPDATE",
    "event_object_table": "line_sessions",
    "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
    "action_timing": "BEFORE",
    "action_orientation": "ROW"
  },
```

```
{
  "trigger_schema": "public",
  "trigger_name": "update_master_list_updated_at",
  "event_manipulation": "UPDATE",
  "event_object_table": "master_list",
  "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
  "action_timing": "BEFORE",
  "action_orientation": "ROW"
},
{
  "trigger_schema": "public",
  "trigger_name": "update_pricing_history_updated_at",
  "event_manipulation": "UPDATE",
  "event_object_table": "pricing_history",
  "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
  "action_timing": "BEFORE",
  "action_orientation": "ROW"
},
{
  "trigger_schema": "public",
  "trigger_name": "update_product_categories_updated_at",
  "event_manipulation": "UPDATE",
  "event_object_table": "product_categories",
  "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
  "action_timing": "BEFORE",
  "action_orientation": "ROW"
},
{
  "trigger_schema": "public",
  "trigger_name": "update_purchase_orders_updated_at",
  "event_manipulation": "UPDATE",
  "event_object_table": "purchase_orders",
  "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
  "action_timing": "BEFORE",
  "action_orientation": "ROW"
},
{
  "trigger_schema": "public",
  "trigger_name": "update_restaurant_people_updated_at",
  "event_manipulation": "UPDATE",
  "event_object_table": "restaurant_people",
  "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
  "action_timing": "BEFORE",
  "action_orientation": "ROW"
},
{
  "trigger_schema": "public",
  "trigger_name": "trigger_update_restaurant_product_preferences_updated_at",
```

```
    "event_manipulation": "UPDATE",
    "event_object_table": "restaurant_product_preferences",
    "action_statement": "EXECUTE FUNCTION
update_restaurant_product_preferences_updated_at()",
    "action_timing": "BEFORE",
    "action_orientation": "ROW"
  },
  {
    "trigger_schema": "public",
    "trigger_name": "update_restaurants_updated_at",
    "event_manipulation": "UPDATE",
    "event_object_table": "restaurants",
    "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
    "action_timing": "BEFORE",
    "action_orientation": "ROW"
  },
  {
    "trigger_schema": "public",
    "trigger_name": "update_supplier_mapped_products_updated_at",
    "event_manipulation": "UPDATE",
    "event_object_table": "supplier_mapped_products",
    "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
    "action_timing": "BEFORE",
    "action_orientation": "ROW"
  },
  {
    "trigger_schema": "public",
    "trigger_name": "update_suppliers_updated_at",
    "event_manipulation": "UPDATE",
    "event_object_table": "suppliers",
    "action_statement": "EXECUTE FUNCTION update_updated_at_column()",
    "action_timing": "BEFORE",
    "action_orientation": "ROW"
  }
]
```


7. Funciones y Procedimientos


```
 [
  {
    "schema": "public",
    "function_name": "_postgis_deprecate",
    "arguments": "oldname text, newname text, version text",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._postgis_deprecate(oldname
text, newname text, version text)\n RETURNS void\n LANGUAGE plpgsql\n IMMUTABLE
```

STRICT COST 500\nAS $function$\nDECLARE\n curver_text text;\nBEGIN\n --\n -- Raises a NOTICE if it was deprecated in this version,\n -- a WARNING if in a previous version (only up to minor version checked)\n --\n\tcurver_text := '3.3.7';\n\tIF pg_catalog.split_part(curver_text,'.',1)::int > pg_catalog.split_part(version,'.',1)::int OR\n\t   (pg_catalog.split_part(curver_text,'.',1) = pg_catalog.split_part(version,'.',1) AND\n\t\t pg_catalog.split_part(curver_text,'.',2) != split_part(version,'.',2))\n\tTHEN\n\t RAISE WARNING '% signature was deprecated in %. Please use %', oldname, version, newname;\n\tELSE\n\t RAISE DEBUG '% signature was deprecated in %. Please use %', oldname, version, newname;\n\tEND IF;\nEND;\n$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_postgis_index_extent",
   "arguments": "tbl regclass, col text",
   "return_type": "box2d",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._postgis_index_extent(tbl regclass, col text)\n RETURNS box2d\n LANGUAGE c\n STABLE STRICT\nAS '$libdir/postgis-3', $function$_postgis_gserialized_index_extent$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_postgis_join_selectivity",
   "arguments": "regclass, text, regclass, text, text DEFAULT '2'::text",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._postgis_join_selectivity(regclass, text, regclass, text, text DEFAULT '2'::text)\n RETURNS double precision\n LANGUAGE c\n PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$_postgis_gserialized_joinsel$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_postgis_pgsql_version",
   "arguments": "",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._postgis_pgsql_version()\n RETURNS text\n LANGUAGE sql\n STABLE\nAS $function$\n\tSELECT CASE WHEN pg_catalog.split_part(s,'.',1)::integer > 9 THEN pg_catalog.split_part(s,'.',1) || '0'\n\tELSE pg_catalog.split_part(s,'.', 1) || pg_catalog.split_part(s,'.', 2) END AS v\n\tFROM pg_catalog.substring(version(), E'PostgreSQL ([0-9\\\\.]+)') AS s;\n$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_postgis_scripts_pgsql_version",
   "arguments": "",
   "return_type": "text",

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public._postgis_scripts_pgsql_version()\n RETURNS text\n LANGUAGE sql\n
IMMUTABLE\nAS $function$SELECT '170'::text AS version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_postgis_selectivity",
    "arguments": "tbl regclass, att_name text, geom geometry, mode text DEFAULT '2'::text",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._postgis_selectivity(tbl regclass,
att_name text, geom geometry, mode text DEFAULT '2'::text)\n RETURNS double
precision\n LANGUAGE c\n PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$_postgis_gserialized_sel$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_postgis_stats",
    "arguments": "tbl regclass, att_name text, text DEFAULT '2'::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._postgis_stats(tbl regclass,
att_name text, text DEFAULT '2'::text)\n RETURNS text\n LANGUAGE c\n PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$_postgis_gserialized_stats$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_3ddfullywithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_3ddfullywithin(geom1
geometry, geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_dfullywithin3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_3ddwithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_3ddwithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_dwithin3d$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "_st_3dintersects",
   "arguments": "geom1 geometry, geom2 geometry",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._st_3dintersects(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_3DIntersects$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_st_asgml",
   "arguments": "integer, geometry, integer, integer, text, text",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._st_asgml(integer, geometry,
integer, integer, text, text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_asGML$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_st_asx3d",
   "arguments": "integer, geometry, integer, integer, text",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._st_asx3d(integer, geometry,
integer, integer, text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_asX3D$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_st_bestsrid",
   "arguments": "geography",
   "return_type": "integer",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public._st_bestsrid(geography)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$geography_bestsrid$function$\n"
 },
 {
   "schema": "public",
   "function_name": "_st_bestsrid",
   "arguments": "geography, geography",
   "return_type": "integer",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public._st_bestsrid(geography,
geography)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$geography_bestsrid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_contains",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_contains(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$contains$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_containsproperly",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_containsproperly(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$containsproperly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_coveredby",
    "arguments": "geog1 geography, geog2 geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_coveredby(geog1 geography,
geog2 geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$geography_coveredby$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_coveredby",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_coveredby(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$coveredby$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_covers",
```

```
    "arguments": "geog1 geography, geog2 geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_covers(geog1 geography,
geog2 geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$geography_covers$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_covers",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_covers(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$covers$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_crosses",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_crosses(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$crosses$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_dfullywithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_dfullywithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_dfullywithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_distancetree",
    "arguments": "geography, geography",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_distancetree(geography,
geography)\n RETURNS double precision\n LANGUAGE sql\n IMMUTABLE STRICT\nAS
$function$SELECT public._ST_DistanceTree($1, $2, 0.0, true)$function$\n"
  },
```

```
  {
    "schema": "public",
    "function_name": "_st_distancetree",
    "arguments": "geography, geography, double precision, boolean",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_distancetree(geography,
geography, double precision, boolean)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$geography_distance_tree$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_distanceuncached",
    "arguments": "geography, geography",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public._st_distanceuncached(geography, geography)\n RETURNS double precision\n
LANGUAGE sql\n IMMUTABLE STRICT\nAS $function$SELECT
public._ST_DistanceUnCached($1, $2, 0.0, true)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_distanceuncached",
    "arguments": "geography, geography, double precision, boolean",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public._st_distanceuncached(geography, geography, double precision, boolean)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$geography_distance_uncached$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_distanceuncached",
    "arguments": "geography, geography, boolean",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public._st_distanceuncached(geography, geography, boolean)\n RETURNS double
precision\n LANGUAGE sql\n IMMUTABLE STRICT\nAS $function$SELECT
public._ST_DistanceUnCached($1, $2, 0.0, $3)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_dwithin",
```

```
    "arguments": "geog1 geography, geog2 geography, tolerance double precision,
use_spheroid boolean DEFAULT true",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_dwithin(geog1 geography,
geog2 geography, tolerance double precision, use_spheroid boolean DEFAULT true)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$geography_dwithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_dwithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_dwithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_dwithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_dwithinuncached",
    "arguments": "geography, geography, double precision, boolean",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_dwithinuncached(geography,
geography, double precision, boolean)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
STRICT COST 10000\nAS '$libdir/postgis-3',
$function$geography_dwithin_uncached$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_dwithinuncached",
    "arguments": "geography, geography, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_dwithinuncached(geography,
geography, double precision)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE\nAS
$function$SELECT $1 OPERATOR(public.&&) public._ST_Expand($2,$3) AND $2
OPERATOR(public.&&) public._ST_Expand($1,$3) AND public._ST_DWithinUnCached($1,
$2, $3, true)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_equals",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_equals(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_Equals$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_expand",
    "arguments": "geography, double precision",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_expand(geography, double
precision)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$geography_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_geomfromgml",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_geomfromgml(text, integer)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS
'$libdir/postgis-3', $function$geom_from_gml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_intersects",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_intersects(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_Intersects$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_linecrossingdirection",
    "arguments": "line1 geometry, line2 geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_linecrossingdirection(line1
geometry, line2 geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_LineCrossingDirection$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "_st_longestline",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_longestline(geom1 geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_longestline2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_maxdistance",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_maxdistance(geom1 geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_maxdistance2d_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_orderingequals",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_orderingequals(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$LWGEOM_same$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_overlaps",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_overlaps(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_pointoutside",
    "arguments": "geography",
    "return_type": "geography",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public._st_pointoutside(geography)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE STRICT\nAS '$libdir/postgis-3', $function$geography_point_outside$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_sortablehash",
    "arguments": "geom geometry",
    "return_type": "bigint",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_sortablehash(geom geometry)\n RETURNS bigint\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$_ST_SortableHash$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_touches",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_touches(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$touches$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_voronoi",
    "arguments": "g1 geometry, clip geometry DEFAULT NULL::geometry, tolerance double precision DEFAULT 0.0, return_polygons boolean DEFAULT true",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_voronoi(g1 geometry, clip geometry DEFAULT NULL::geometry, tolerance double precision DEFAULT 0.0, return_polygons boolean DEFAULT true)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 10000\nAS '$libdir/postgis-3', $function$ST_Voronoi$function$\n"
  },
  {
    "schema": "public",
    "function_name": "_st_within",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public._st_within(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$SELECT public._ST_Contains($2,$1)$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "addauth",
    "arguments": "text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.addauth(text)\n RETURNS
boolean\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\tlockid alias for $1;\n\tokay
boolean;\n\tmyrec record;\nBEGIN\n\t-- check to see if table exists\n\t--  if not, CREATE
TEMP TABLE mylock (transid xid, lockcode text)\n\tokay := 'f';\n\tFOR myrec IN SELECT *
FROM pg_class WHERE relname = 'temp_lock_have_table' LOOP\n\t\tokay := 't';\n\tEND
LOOP;\n\tIF (okay <> 't') THEN\n\t\tCREATE TEMP TABLE temp_lock_have_table (transid
xid, lockcode text);\n\t\t\t-- this will only work from pgsql7.4 up\n\t\t\t-- ON COMMIT DELETE
ROWS;\n\tEND IF;\n\n\t--  INSERT INTO mylock VALUES ( $1)\n--\tEXECUTE 'INSERT
INTO temp_lock_have_table VALUES ( '||\n--\t\tquote_literal(getTransactionID()) || ','
||\n--\t\tquote_literal(lockid) ||')';\n\n\tINSERT INTO temp_lock_have_table VALUES
(getTransactionID(), lockid);\n\n\tRETURN true::boolean;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "addgeometrycolumn",
    "arguments": "schema_name character varying, table_name character varying,
column_name character varying, new_srid integer, new_type character varying, new_dim
integer, use_typmod boolean DEFAULT true",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.addgeometrycolumn(schema_name character varying, table_name character varying,
column_name character varying, new_srid integer, new_type character varying, new_dim
integer, use_typmod boolean DEFAULT true)\n RETURNS text\n LANGUAGE plpgsql\n
STABLE STRICT\nAS $function$\nDECLARE\n\tret  text;\nBEGIN\n\tSELECT
public.AddGeometryColumn('',$1,$2,$3,$4,$5,$6,$7) into ret;\n\tRETURN
ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "addgeometrycolumn",
    "arguments": "catalog_name character varying, schema_name character varying,
table_name character varying, column_name character varying, new_srid_in integer,
new_type character varying, new_dim integer, use_typmod boolean DEFAULT true",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.addgeometrycolumn(catalog_name character varying, schema_name character
varying, table_name character varying, column_name character varying, new_srid_in
integer, new_type character varying, new_dim integer, use_typmod boolean DEFAULT
true)\n RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\trec
RECORD;\n\tsr varchar;\n\treal_schema name;\n\tsql text;\n\tnew_srid
integer;\n\nBEGIN\n\n\t-- Verify geometry type\n\tIF
```

(postgis_type_name(new_type,new_dim) IS NULL)\n\tTHEN\n\t\tRAISE EXCEPTION 'Invalid type name \"%(%)\" - valid ones are:\n\tPOINT, MULTIPOINT,\n\tLINESTRING, MULTILINESTRING,\n\tPOLYGON, MULTIPOLYGON,\n\tCIRCULARSTRING, COMPOUNDCURVE, MULTICURVE,\n\tCURVEPOLYGON, MULTISURFACE,\n\tGEOMETRY, GEOMETRYCOLLECTION,\n\tPOINTM, MULTIPOINTM,\n\tLINESTRINGM, MULTILINESTRINGM,\n\tPOLYGONM, MULTIPOLYGONM,\n\tCIRCULARSTRINGM, COMPOUNDCURVEM, MULTICURVEM\n\tCURVEPOLYGONM, MULTISURFACEM, TRIANGLE, TRIANGLEM,\n\tPOLYHEDRALSURFACE, POLYHEDRALSURFACEM, TIN, TINM\n\tor GEOMETRYCOLLECTIONM', new_type, new_dim;\n\t\tRETURN 'fail';\n\tEND IF;\n\n\t-- Verify dimension\n\tIF ((new_dim >4) OR (new_dim <2)) THEN\n\t\tRAISE EXCEPTION 'invalid dimension';\n\t\tRETURN 'fail';\n\tEND IF;\n\n\tIF ((new_type LIKE '%M') AND (new_dim!=3)) THEN\n\t\tRAISE EXCEPTION 'TypeM needs 3 dimensions';\n\t\tRETURN 'fail';\n\tEND IF;\n\n\t-- Verify SRID\n\tIF (new_srid_in > 0) THEN\n\t\tIF new_srid_in > 998999 THEN\n\t\t\tRAISE EXCEPTION 'AddGeometryColumn() - SRID must be <= %', 998999;\n\t\tEND IF;\n\t\tnew_srid := new_srid_in;\n\t\tSELECT SRID INTO sr FROM spatial_ref_sys WHERE SRID = new_srid;\n\t\tIF NOT FOUND THEN\n\t\t\tRAISE EXCEPTION 'AddGeometryColumn() - invalid SRID';\n\t\t\tRETURN 'fail';\n\t\tEND IF;\n\tELSE\n\t\tnew_srid := public.ST_SRID('POINT EMPTY'::public.geometry);\n\t\tIF (new_srid_in != new_srid) THEN\n\t\t\tRAISE NOTICE 'SRID value % converted to the officially unknown SRID value %', new_srid_in, new_srid;\n\t\tEND IF;\n\tEND IF;\n\n\t-- Verify schema\n\tIF (schema_name IS NOT NULL AND schema_name != '') THEN\n\t\tsql := 'SELECT nspname FROM pg_namespace ' ||\n\t\t\t'WHERE text(nspname) = ' || quote_literal(schema_name) ||\n\t\t\t'LIMIT 1';\n\t\tRAISE DEBUG '%', sql;\n\t\tEXECUTE sql INTO real_schema;\n\n\tIF (real_schema IS NULL) THEN\n\t\t\tRAISE EXCEPTION 'Schema % is not a valid schemaname', quote_literal(schema_name);\n\t\t\tRETURN 'fail';\n\t\tEND IF;\n\tEND IF;\n\n\tIF (real_schema IS NULL) THEN\n\t\tRAISE DEBUG 'Detecting schema';\n\t\tsql := 'SELECT n.nspname AS schemaname ' ||\n\t\t\t'FROM pg_catalog.pg_class c ' ||\n\t\t\t 'JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace ' ||\n\t\t\t'WHERE c.relkind = ' || quote_literal('r') ||\n\t\t\t' AND n.nspname NOT IN (' || quote_literal('pg_catalog') || ', ' || quote_literal('pg_toast') || ')' ||\n\t\t\t' AND pg_catalog.pg_table_is_visible(c.oid)' ||\n\t\t\t' AND c.relname = ' || quote_literal(table_name);\n\t\tRAISE DEBUG '%', sql;\n\t\tEXECUTE sql INTO real_schema;\n\n\t\tIF (real_schema IS NULL) THEN\n\t\t\tRAISE EXCEPTION 'Table % does not occur in the search_path', quote_literal(table_name);\n\t\t\tRETURN 'fail';\n\t\tEND IF;\n\tEND IF;\n\n\t-- Add geometry column to table\n\tIF use_typmod THEN\n\t\t sql := 'ALTER TABLE ' ||\n\t\t\tquote_ident(real_schema) || '.' || quote_ident(table_name)\n\t\t\t|| ' ADD COLUMN ' || quote_ident(column_name) ||\n\t\t\t' geometry(' || public.postgis_type_name(new_type, new_dim) || ', ' || new_srid::text || ')';\n\t\tRAISE DEBUG '%', sql;\n\tELSE\n\t\tsql := 'ALTER TABLE ' ||\n\t\t\tquote_ident(real_schema) || '.' || quote_ident(table_name)\n\t\t\t|| ' ADD COLUMN ' || quote_ident(column_name) ||\n\t\t\t' geometry ';\n\t\tRAISE DEBUG '%', sql;\n\tEND IF;\n\tEXECUTE sql;\n\n\tIF NOT use_typmod THEN\n\t\t-- Add table CHECKs\n\t\tsql := 'ALTER TABLE ' ||\n\t\t\tquote_ident(real_schema) || '.' || quote_ident(table_name)\n\t\t\t|| ' ADD CONSTRAINT '\n\t\t\t|| quote_ident('enforce_srid_' || column_name)\n\t\t\t|| ' CHECK (st_srid(' || quote_ident(column_name) ||\n\t\t\t') = ' || new_srid::text || ')' ;\n\t\tRAISE DEBUG '%', sql;\n\t\tEXECUTE sql;\n\n\t\tsql := 'ALTER TABLE ' ||\n\t\t\tquote_ident(real_schema) || '.' || quote_ident(table_name)\n\t\t\t|| ' ADD CONSTRAINT '\n\t\t\t||

quote_ident('enforce_dims_' || column_name)\n\t\t\t|| ' CHECK (st_ndims(' || quote_ident(column_name) ||\n\t\t\t') = ' || new_dim::text || ')' ;\n\t\tRAISE DEBUG '%', sql;\n\t\tEXECUTE sql;\n\n\t\tIF (NOT (new_type = 'GEOMETRY')) THEN\n\t\t\tsql := 'ALTER TABLE ' ||\n\t\t\t\tquote_ident(real_schema) || '.' || quote_ident(table_name) || ' ADD CONSTRAINT ' ||\n\t\t\t\tquote_ident('enforce_geotype_' || column_name) ||\n\t\t\t\t' CHECK (GeometryType(' ||\n\t\t\t\tquote_ident(column_name) || ')=' ||\n\t\t\t\tquote_literal(new_type) || ' OR (' ||\n\t\t\t\tquote_ident(column_name) || ') is null)';\n\t\t\tRAISE DEBUG '%', sql;\n\t\t\tEXECUTE sql;\n\t\tEND IF;\n\tEND IF;\n\n\tRETURN\n\t\treal_schema || '.' ||\n\t\ttable_name || '.' || column_name ||\n\t\t' SRID:' || new_srid::text ||\n\t\t' TYPE:' || new_type ||\n\t\t' DIMS:' || new_dim::text || ' ';\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "addgeometrycolumn",
    "arguments": "table_name character varying, column_name character varying, new_srid integer, new_type character varying, new_dim integer, use_typmod boolean DEFAULT true",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.addgeometrycolumn(table_name character varying, column_name character varying, new_srid integer, new_type character varying, new_dim integer, use_typmod boolean DEFAULT true)\n RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tret text;\nBEGIN\n\tSELECT public.AddGeometryColumn('','',$1,$2,$3,$4,$5, $6) into ret;\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_halfvec",
    "arguments": "double precision[], integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_halfvec(double precision[], integer, boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$array_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_halfvec",
    "arguments": "integer[], integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_halfvec(integer[], integer, boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$array_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_halfvec",

```
    "arguments": "real[], integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_halfvec(real[], integer,
boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_halfvec",
    "arguments": "numeric[], integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_halfvec(numeric[],
integer, boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_sparsevec",
    "arguments": "integer[], integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_sparsevec(integer[],
integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_sparsevec",
    "arguments": "double precision[], integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_sparsevec(double
precision[], integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$array_to_sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_sparsevec",
    "arguments": "numeric[], integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_sparsevec(numeric[],
integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_sparsevec$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "array_to_sparsevec",
    "arguments": "real[], integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_sparsevec(real[],
integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_vector",
    "arguments": "numeric[], integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_vector(numeric[],
integer, boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_vector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_vector",
    "arguments": "integer[], integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_vector(integer[], integer,
boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_vector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_vector",
    "arguments": "real[], integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_vector(real[], integer,
boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$array_to_vector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "array_to_vector",
    "arguments": "double precision[], integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.array_to_vector(double
precision[], integer, boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$array_to_vector$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "binary_quantize",
    "arguments": "halfvec",
    "return_type": "bit",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.binary_quantize(halfvec)\n RETURNS bit\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_binary_quantize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "binary_quantize",
    "arguments": "vector",
    "return_type": "bit",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.binary_quantize(vector)\n RETURNS bit\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$binary_quantize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box",
    "arguments": "box3d",
    "return_type": "box",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box(box3d)\n RETURNS box\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$BOX3D_to_BOX$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box",
    "arguments": "geometry",
    "return_type": "box",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box(geometry)\n RETURNS box\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_to_BOX$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2d",
    "arguments": "geometry",
    "return_type": "box2d",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.box2d(geometry)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_to_BOX2D$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2d",
    "arguments": "box3d",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box2d(box3d)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$BOX3D_to_BOX2D$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2d_in",
    "arguments": "cstring",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box2d_in(cstring)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$BOX2D_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2d_out",
    "arguments": "box2d",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box2d_out(box2d)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$BOX2D_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2df_in",
    "arguments": "cstring",
    "return_type": "box2df",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box2df_in(cstring)\n RETURNS box2df\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$box2df_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box2df_out",
    "arguments": "box2df",
```

```
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box2df_out(box2df)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$box2df_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box3d",
    "arguments": "box2d",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box3d(box2d)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$BOX2D_to_BOX3D$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box3d",
    "arguments": "geometry",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box3d(geometry)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_to_BOX3D$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box3d_in",
    "arguments": "cstring",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box3d_in(cstring)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$BOX3D_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "box3d_out",
    "arguments": "box3d",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box3d_out(box3d)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$BOX3D_out$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "box3dtobox",
    "arguments": "box3d",
    "return_type": "box",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.box3dtobox(box3d)\n RETURNS box\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$BOX3D_to_BOX$function$\n"
  },
  {
    "schema": "public",
    "function_name": "bytea",
    "arguments": "geometry",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.bytea(geometry)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_to_bytea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "bytea",
    "arguments": "geography",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.bytea(geography)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_to_bytea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "checkauth",
    "arguments": "text, text, text",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.checkauth(text, text, text)\n RETURNS integer\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\tschema text;\nBEGIN\n\tIF NOT LongTransactionsEnabled() THEN\n\t\tRAISE EXCEPTION 'Long transaction support disabled, use EnableLongTransaction() to enable.';\n\tEND IF;\n\n\tif ( $1 != '' ) THEN\n\t\tschema = $1;\n\tELSE\n\t\tSELECT current_schema() into schema;\n\tEND IF;\n\n\t-- TODO: check for an already existing trigger ?\n\n\tEXECUTE 'CREATE TRIGGER check_auth BEFORE UPDATE OR DELETE ON '\n\t\t|| quote_ident(schema) || '.' || quote_ident($2)\n\t\t||' FOR EACH ROW EXECUTE PROCEDURE CheckAuthTrigger('\n\t\t|| quote_literal($3) || ')';\n\n\tRETURN 0;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "checkauth",
    "arguments": "text, text",
```

```
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.checkauth(text, text)\n RETURNS integer\n LANGUAGE sql\nAS $function$ SELECT CheckAuth('', $1, $2) $function$\n"
  },
  {
    "schema": "public",
    "function_name": "checkauthtrigger",
    "arguments": "",
    "return_type": "trigger",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.checkauthtrigger()\n RETURNS trigger\n LANGUAGE c\nAS '$libdir/postgis-3', $function$check_authorization$function$\n"
  },
  {
    "schema": "public",
    "function_name": "contains_2d",
    "arguments": "box2df, geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.contains_2d(box2df, geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_contains_box2df_geom_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "contains_2d",
    "arguments": "geometry, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.contains_2d(geometry, box2df)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 1\nAS $function$SELECT $2 OPERATOR(public.@) $1;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "contains_2d",
    "arguments": "box2df, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.contains_2d(box2df, box2df)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_contains_box2df_box2df_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "cosine_distance",
    "arguments": "sparsevec, sparsevec",
```

```
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.cosine_distance(sparsevec,
sparsevec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_cosine_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "cosine_distance",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.cosine_distance(halfvec,
halfvec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec_cosine_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "cosine_distance",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.cosine_distance(vector, vector)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$cosine_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "disablelongtransactions",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.disablelongtransactions()\n
RETURNS text\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\trec
RECORD;\n\nBEGIN\n\n\t--\n\t-- Drop all triggers applied by CheckAuth()\n\t--\n\tFOR rec
IN\n\t\tSELECT c.relname, t.tgname, t.tgargs FROM pg_trigger t, pg_class c, pg_proc
p\n\t\tWHERE p.proname = 'checkauthtrigger' and t.tgfoid = p.oid and t.tgrelid =
c.oid\n\tLOOP\n\t\tEXECUTE 'DROP TRIGGER ' || quote_ident(rec.tgname) ||\n\t\t\t' ON ' ||
quote_ident(rec.relname);\n\tEND LOOP;\n\n\t--\n\t-- Drop the authorization_table
table\n\t--\n\tFOR rec IN SELECT * FROM pg_class WHERE relname = 'authorization_table'
LOOP\n\t\tDROP TABLE authorization_table;\n\tEND LOOP;\n\n\t--\n\t-- Drop the
authorized_tables view\n\t--\n\tFOR rec IN SELECT * FROM pg_class WHERE relname =
'authorized_tables' LOOP\n\t\tDROP VIEW authorized_tables;\n\tEND LOOP;\n\n\tRETURN
'Long transactions support disabled';\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "dropgeometrycolumn",
```

```
    "arguments": "table_name character varying, column_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.dropgeometrycolumn(table_name
character varying, column_name character varying)\n RETURNS text\n LANGUAGE
plpgsql\n STRICT\nAS $function$\nDECLARE\n\tret text;\nBEGIN\n\tSELECT
public.DropGeometryColumn('','',$1,$2) into ret;\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "dropgeometrycolumn",
    "arguments": "schema_name character varying, table_name character varying,
column_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.dropgeometrycolumn(schema_name character varying, table_name character
varying, column_name character varying)\n RETURNS text\n LANGUAGE plpgsql\n
STRICT\nAS $function$\nDECLARE\n\tret text;\nBEGIN\n\tSELECT
public.DropGeometryColumn('',$1,$2,$3) into ret;\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "dropgeometrycolumn",
    "arguments": "catalog_name character varying, schema_name character varying,
table_name character varying, column_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.dropgeometrycolumn(catalog_name character varying, schema_name character
varying, table_name character varying, column_name character varying)\n RETURNS text\n
LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tmyrec RECORD;\n\tokay
boolean;\n\treal_schema name;\n\nBEGIN\n\n\t-- Find, check or fix schema_name\n\tIF (
schema_name != '' ) THEN\n\t\tokay = false;\n\n\t\tFOR myrec IN SELECT nspname FROM
pg_namespace WHERE text(nspname) = schema_name LOOP\n\t\t\tokay := true;\n\t\tEND
LOOP;\n\n\t\tIF ( okay <>  true ) THEN\n\t\t\tRAISE NOTICE 'Invalid schema name - using
current_schema()';\n\t\t\tSELECT current_schema() into
real_schema;\n\t\tELSE\n\t\t\treal_schema = schema_name;\n\t\tEND
IF;\n\tELSE\n\t\tSELECT current_schema() into real_schema;\n\tEND IF;\n\n\t-- Find out if
the column is in the geometry_columns table\n\tokay = false;\n\tFOR myrec IN SELECT *
from public.geometry_columns where f_table_schema = text(real_schema) and
f_table_name = table_name and f_geometry_column = column_name LOOP\n\t\tokay :=
true;\n\tEND LOOP;\n\tIF (okay <> true) THEN\n\t\tRAISE EXCEPTION 'column not found in
geometry_columns table';\n\t\tRETURN false;\n\tEND IF;\n\n\t-- Remove table
column\n\tEXECUTE 'ALTER TABLE ' || quote_ident(real_schema) || '.'
||\n\t\tquote_ident(table_name) || ' DROP COLUMN '
||\n\t\tquote_ident(column_name);\n\n\tRETURN real_schema || '.' || table_name || '.' ||
column_name ||' effectively removed.';\n\nEND;\n$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "dropgeometrytable",
    "arguments": "table_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.dropgeometrytable(table_name
character varying)\n RETURNS text\n LANGUAGE sql\n STRICT\nAS $function$ SELECT
public.DropGeometryTable('','',$1) $function$\n"
  },
  {
    "schema": "public",
    "function_name": "dropgeometrytable",
    "arguments": "catalog_name character varying, schema_name character varying,
table_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.dropgeometrytable(catalog_name
character varying, schema_name character varying, table_name character varying)\n
RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS
$function$\nDECLARE\n\treal_schema name;\n\nBEGIN\n\n\tIF ( schema_name = '' )
THEN\n\t\tSELECT current_schema() into real_schema;\n\tELSE\n\t\treal_schema =
schema_name;\n\tEND IF;\n\n\t-- TODO: Should we warn if table doesn't exist probably
instead just saying dropped\n\t-- Remove table\n\tEXECUTE 'DROP TABLE IF EXISTS
'\n\t\t|| quote_ident(real_schema) || '.' ||\n\t\tquote_ident(table_name) || '
RESTRICT';\n\n\tRETURN\n\t\treal_schema || '.' ||\n\t\ttable_name ||'
dropped.';\n\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "dropgeometrytable",
    "arguments": "schema_name character varying, table_name character varying",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.dropgeometrytable(schema_name character varying, table_name character varying)\n
RETURNS text\n LANGUAGE sql\n STRICT\nAS $function$ SELECT
public.DropGeometryTable('',$1,$2) $function$\n"
  },
  {
    "schema": "public",
    "function_name": "enablelongtransactions",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.enablelongtransactions()\n
RETURNS text\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\t\"query\" text;\n\texists
```

bool;\n\trec RECORD;\n\nBEGIN\n\n\texists = 'f';\n\tFOR rec IN SELECT * FROM pg_class WHERE relname = 'authorization_table'\n\tLOOP\n\t\texists = 't';\n\tEND LOOP;\n\n\tIF NOT exists\n\tTHEN\n\t\t\"query\" = 'CREATE TABLE authorization_table (\n\t\t\ttoid oid, -- table oid\n\t\t\trid text, -- row id\n\t\t\texpires timestamp,\n\t\t\tauthid text\n\t\t)';\n\t\tEXECUTE \"query\";\n\tEND IF;\n\n\texists = 'f';\n\tFOR rec IN SELECT * FROM pg_class WHERE relname = 'authorized_tables'\n\tLOOP\n\t\texists = 't';\n\tEND LOOP;\n\n\tIF NOT exists THEN\n\t\t\"query\" = 'CREATE VIEW authorized_tables AS ' ||\n\t\t'SELECT ' ||\n\t\t'n.nspname as schema, ' ||\n\t\t'c.relname as table, trim(' ||\n\t\t\tquote_literal(chr(92) || '000') ||\n\t\t' from t.tgargs) as id_column ' ||\n\t\t'FROM pg_trigger t, pg_class c, pg_proc p ' ||\n\t\t', pg_namespace n ' ||\n\t\t'WHERE p.proname = ' || quote_literal('checkauthtrigger') ||\n\t\t' AND c.relnamespace = n.oid' ||\n\t\t' AND t.tgfoid = p.oid and t.tgrelid = c.oid';\n\t\tEXECUTE \"query\";\n\tEND IF;\n\n\tRETURN 'Long transactions support enabled';\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "equals",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.equals(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_Equals$function$\n"
  },
  {
    "schema": "public",
    "function_name": "find_srid",
    "arguments": "character varying, character varying, character varying",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.find_srid(character varying, character varying, character varying)\n RETURNS integer\n LANGUAGE plpgsql\n STABLE PARALLEL SAFE STRICT\nAS $function$\nDECLARE\n\tschem varchar = $1;\n\ttabl varchar = $2;\n\tsr int4;\nBEGIN\n-- if the table contains a . and the schema is empty\n-- split the table into a schema and a table\n-- otherwise drop through to default behavior\n\tIF (schem = '' and strpos(tabl,'.') > 0) THEN\n\t schem = substr(tabl,1,strpos(tabl,'.')-1);\n\t tabl = substr(tabl,length(schem)+2);\n\tEND IF;\n\n\tselect SRID into sr from public.geometry_columns where (f_table_schema = schem or schem = '') and f_table_name = tabl and f_geometry_column = $3;\n\tIF NOT FOUND THEN\n\t   RAISE EXCEPTION 'find_srid() - could not find the corresponding SRID - is the geometry registered in the GEOMETRY_COLUMNS table? Is there an uppercase/lowercase mismatch?';\n\tEND IF;\n\treturn sr;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geog_brin_inclusion_add_value",
    "arguments": "internal, internal, internal, internal",
    "return_type": "boolean",

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geog_brin_inclusion_add_value(internal, internal, internal, internal)\n RETURNS
boolean\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$geog_brin_inclusion_add_value$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography",
    "arguments": "bytea",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography(bytea)\n RETURNS
geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geography_from_binary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography",
    "arguments": "geography, integer, boolean",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography(geography, integer,
boolean)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_enforce_typmod$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography",
    "arguments": "geometry",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography(geometry)\n
RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geography_from_geometry$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_analyze",
    "arguments": "internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_analyze(internal)\n
RETURNS boolean\n LANGUAGE c\n STRICT\nAS '$libdir/postgis-3',
$function$gserialized_analyze_nd$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "geography_cmp",
    "arguments": "geography, geography",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_cmp(geography,
geography)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_cmp$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_distance_knn",
    "arguments": "geography, geography",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_distance_knn(geography, geography)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 100\nAS '$libdir/postgis-3',
$function$geography_distance_knn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_eq",
    "arguments": "geography, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_eq(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_eq$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_ge",
    "arguments": "geography, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_ge(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_ge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_compress",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_gist_compress(internal)\n RETURNS internal\n LANGUAGE c\nAS
'$libdir/postgis-3', $function$gserialized_gist_compress$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "geography_gist_consistent",
    "arguments": "internal, geography, integer",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_gist_consistent(internal, geography, integer)\n RETURNS boolean\n
LANGUAGE c\nAS '$libdir/postgis-3', $function$gserialized_gist_consistent$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_decompress",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_gist_decompress(internal)\n RETURNS internal\n LANGUAGE c\nAS
'$libdir/postgis-3', $function$gserialized_gist_decompress$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_distance",
    "arguments": "internal, geography, integer",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_gist_distance(internal, geography, integer)\n RETURNS double
precision\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$gserialized_gist_geog_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_penalty",
    "arguments": "internal, internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_gist_penalty(internal,
internal, internal)\n RETURNS internal\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$gserialized_gist_penalty$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_picksplit",
    "arguments": "internal, internal",
    "return_type": "internal",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.geography_gist_picksplit(internal,
internal)\n RETURNS internal\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$gserialized_gist_picksplit$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_same",
    "arguments": "box2d, box2d, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_gist_same(box2d,
box2d, internal)\n RETURNS internal\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$gserialized_gist_same$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gist_union",
    "arguments": "bytea, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_gist_union(bytea,
internal)\n RETURNS internal\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$gserialized_gist_union$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_gt",
    "arguments": "geography, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_gt(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_gt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_in",
    "arguments": "cstring, oid, integer",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_in(cstring, oid,
integer)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_le",
    "arguments": "geography, geography",
```

```
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_le(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_le$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_lt",
    "arguments": "geography, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_lt(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_lt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_out",
    "arguments": "geography",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_out(geography)\n
RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geography_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_overlaps",
    "arguments": "geography, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_overlaps(geography,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_recv",
    "arguments": "internal, oid, integer",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_recv(internal, oid,
integer)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geography_recv$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "geography_send",
    "arguments": "geography",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_send(geography)\n
RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geography_send$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_choose_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_choose_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_choose_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_compress_nd",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_compress_nd(internal)\n RETURNS internal\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_compress_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_config_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_config_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_config_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_inner_consistent_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_inner_consistent_nd(internal, internal)\n RETURNS void\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_inner_consistent_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_leaf_consistent_nd",
    "arguments": "internal, internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_leaf_consistent_nd(internal, internal)\n RETURNS boolean\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_leaf_consistent_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_spgist_picksplit_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_spgist_picksplit_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_picksplit_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_typmod_in",
    "arguments": "cstring[]",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geography_typmod_in(cstring[])\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geography_typmod_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geography_typmod_out",
    "arguments": "integer",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geography_typmod_out(integer)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$postgis_typmod_out$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "geom2d_brin_inclusion_add_value",
   "arguments": "internal, internal, internal, internal",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geom2d_brin_inclusion_add_value(internal, internal, internal, internal)\n RETURNS
boolean\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$geom2d_brin_inclusion_add_value$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geom3d_brin_inclusion_add_value",
   "arguments": "internal, internal, internal, internal",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geom3d_brin_inclusion_add_value(internal, internal, internal, internal)\n RETURNS
boolean\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$geom3d_brin_inclusion_add_value$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geom4d_brin_inclusion_add_value",
   "arguments": "internal, internal, internal, internal",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geom4d_brin_inclusion_add_value(internal, internal, internal, internal)\n RETURNS
boolean\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$geom4d_brin_inclusion_add_value$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry",
   "arguments": "polygon",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.geometry(polygon)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$polygon_to_geometry$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry",
   "arguments": "geometry, integer, boolean",
   "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(geometry, integer,
boolean)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$geometry_enforce_typmod$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(bytea)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_from_bytea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(text)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$parse_WKT_lwgeom$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "box3d",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(box3d)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$BOX3D_to_LWGEOM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "box2d",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(box2d)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$BOX2D_to_LWGEOM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
```

```
    "arguments": "point",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(point)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$point_to_geometry$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "geography",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(geography)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geometry_from_geography$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry",
    "arguments": "path",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry(path)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$path_to_geometry$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_above",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_above(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_above_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_analyze",
    "arguments": "internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_analyze(internal)\n
RETURNS boolean\n LANGUAGE c\n STRICT\nAS '$libdir/postgis-3',
$function$gserialized_analyze_nd$function$\n"
  },
```

```
  {
    "schema": "public",
    "function_name": "geometry_below",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_below(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_below_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_cmp",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_cmp(geom1 geometry,
geom2 geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$lwgeom_cmp$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_contained_3d",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_contained_3d(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_contained_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_contains",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_contains(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_contains_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_contains_3d",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_contains_3d(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_contains_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_contains_nd",
    "arguments": "geometry, geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_contains_nd(geometry,
geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_contains$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_distance_box",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_distance_box(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_distance_box_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_distance_centroid",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_distance_centroid(geom1 geometry, geom2 geometry)\n RETURNS double
precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$ST_Distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_distance_centroid_nd",
    "arguments": "geometry, geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_distance_centroid_nd(geometry, geometry)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_distance_nd$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "geometry_distance_cpa",
    "arguments": "geometry, geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_distance_cpa(geometry, geometry)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_DistanceCPA$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_eq",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_eq(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$lwgeom_eq$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_ge",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_ge(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$lwgeom_ge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_compress_2d",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_compress_2d(internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_compress_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_compress_nd",
    "arguments": "internal",
    "return_type": "internal",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_compress_nd(internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_compress$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_consistent_2d",
    "arguments": "internal, geometry, integer",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_consistent_2d(internal, geometry, integer)\n RETURNS boolean\n
LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_consistent_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_consistent_nd",
    "arguments": "internal, geometry, integer",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_consistent_nd(internal, geometry, integer)\n RETURNS boolean\n
LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_consistent$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_decompress_2d",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_decompress_2d(internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_decompress_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_decompress_nd",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_decompress_nd(internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_decompress$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_distance_2d",
    "arguments": "internal, geometry, integer",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_distance_2d(internal, geometry, integer)\n RETURNS double
precision\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_distance_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_distance_nd",
    "arguments": "internal, geometry, integer",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_distance_nd(internal, geometry, integer)\n RETURNS double
precision\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_penalty_2d",
    "arguments": "internal, internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_penalty_2d(internal, internal, internal)\n RETURNS internal\n
LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_penalty_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_penalty_nd",
    "arguments": "internal, internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_penalty_nd(internal, internal, internal)\n RETURNS internal\n
LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_penalty$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_picksplit_2d",
```

```
    "arguments": "internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_picksplit_2d(internal, internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_picksplit_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_picksplit_nd",
    "arguments": "internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_picksplit_nd(internal, internal)\n RETURNS internal\n LANGUAGE c\n
PARALLEL SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_picksplit$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_same_2d",
    "arguments": "geom1 geometry, geom2 geometry, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_gist_same_2d(geom1
geometry, geom2 geometry, internal)\n RETURNS internal\n LANGUAGE c\n PARALLEL
SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_same_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_same_nd",
    "arguments": "geometry, geometry, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_same_nd(geometry, geometry, internal)\n RETURNS internal\n
LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_same$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_sortsupport_2d",
    "arguments": "internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_gist_sortsupport_2d(internal)\n RETURNS void\n LANGUAGE c\n
STRICT\nAS '$libdir/postgis-3', $function$gserialized_gist_sortsupport_2d$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_union_2d",
    "arguments": "bytea, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_gist_union_2d(bytea,
internal)\n RETURNS internal\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_union_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gist_union_nd",
    "arguments": "bytea, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_gist_union_nd(bytea,
internal)\n RETURNS internal\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_union$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_gt",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_gt(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$lwgeom_gt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_hash",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_hash(geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$lwgeom_hash$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_in",
    "arguments": "cstring",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_in(cstring)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_le",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_le(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$lwgeom_le$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_left",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_left(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_left_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_lt",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_lt(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$lwgeom_lt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_out",
    "arguments": "geometry",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_out(geometry)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overabove",
    "arguments": "geom1 geometry, geom2 geometry",
```

```
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overabove(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overabove_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overbelow",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overbelow(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overbelow_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overlaps",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overlaps(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overlaps_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overlaps_3d",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overlaps_3d(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overlaps_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overlaps_nd",
    "arguments": "geometry, geometry",
    "return_type": "boolean",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overlaps_nd(geometry,
geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overleft",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overleft(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overleft_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_overright",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_overright(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_overright_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_recv",
    "arguments": "internal",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_recv(internal)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_recv$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_right",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_right(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_right_2d$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "geometry_same",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_same(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_same_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_same_3d",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_same_3d(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_same_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_same_nd",
    "arguments": "geometry, geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_same_nd(geometry,
geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_same$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_send",
    "arguments": "geometry",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_send(geometry)\n
RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_send$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_sortsupport",
    "arguments": "internal",
    "return_type": "void",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_sortsupport(internal)\n RETURNS void\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$lwgeom_sortsupport$function$\n"
   },
   {
    "schema": "public",
    "function_name": "geometry_spgist_choose_2d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_spgist_choose_2d(internal, internal)\n RETURNS void\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_spgist_choose_2d$function$\n"
   },
   {
    "schema": "public",
    "function_name": "geometry_spgist_choose_3d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_spgist_choose_3d(internal, internal)\n RETURNS void\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_spgist_choose_3d$function$\n"
   },
   {
    "schema": "public",
    "function_name": "geometry_spgist_choose_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_spgist_choose_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_spgist_choose_nd$function$\n"
   },
   {
    "schema": "public",
    "function_name": "geometry_spgist_compress_2d",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_spgist_compress_2d(internal)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gserialized_spgist_compress_2d$function$\n"
   },
```

```
 {
   "schema": "public",
   "function_name": "geometry_spgist_compress_3d",
   "arguments": "internal",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_compress_3d(internal)\n RETURNS internal\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_compress_3d$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry_spgist_compress_nd",
   "arguments": "internal",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_compress_nd(internal)\n RETURNS internal\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_compress_nd$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry_spgist_config_2d",
   "arguments": "internal, internal",
   "return_type": "void",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_config_2d(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_config_2d$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry_spgist_config_3d",
   "arguments": "internal, internal",
   "return_type": "void",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_config_3d(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_config_3d$function$\n"
 },
 {
   "schema": "public",
   "function_name": "geometry_spgist_config_nd",
   "arguments": "internal, internal",
```

```
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_config_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_config_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_inner_consistent_2d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_inner_consistent_2d(internal, internal)\n RETURNS void\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_inner_consistent_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_inner_consistent_3d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_inner_consistent_3d(internal, internal)\n RETURNS void\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_inner_consistent_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_inner_consistent_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_inner_consistent_nd(internal, internal)\n RETURNS void\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_inner_consistent_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_leaf_consistent_2d",
    "arguments": "internal, internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_leaf_consistent_2d(internal, internal)\n RETURNS boolean\n
```

LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_leaf_consistent_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_leaf_consistent_3d",
    "arguments": "internal, internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_leaf_consistent_3d(internal, internal)\n RETURNS boolean\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_leaf_consistent_3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_leaf_consistent_nd",
    "arguments": "internal, internal",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_leaf_consistent_nd(internal, internal)\n RETURNS boolean\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_leaf_consistent_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_picksplit_2d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_picksplit_2d(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_picksplit_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_spgist_picksplit_3d",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_picksplit_3d(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_picksplit_3d$function$\n"
  },
  {

```
    "schema": "public",
    "function_name": "geometry_spgist_picksplit_nd",
    "arguments": "internal, internal",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.geometry_spgist_picksplit_nd(internal, internal)\n RETURNS void\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_spgist_picksplit_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_typmod_in",
    "arguments": "cstring[]",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_typmod_in(cstring[])\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geometry_typmod_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_typmod_out",
    "arguments": "integer",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_typmod_out(integer)\n
RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$postgis_typmod_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_within",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_within(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_within_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometry_within_nd",
    "arguments": "geometry, geometry",
    "return_type": "boolean",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.geometry_within_nd(geometry,
geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_within$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometrytype",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometrytype(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_getTYPE$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geometrytype",
    "arguments": "geography",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geometrytype(geography)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_getTYPE$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geomfromewkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geomfromewkb(bytea)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOMFromEWKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "geomfromewkt",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.geomfromewkt(text)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$parse_WKT_lwgeom$function$\n"
  },
  {
    "schema": "public",
    "function_name": "get_proj4_from_srid",
    "arguments": "integer",
```

```
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.get_proj4_from_srid(integer)\n RETURNS text\n LANGUAGE plpgsql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$\n\tBEGIN\n\tRETURN proj4text::text FROM public.spatial_ref_sys WHERE srid= $1;\n\tEND;\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gettransactionid",
    "arguments": "",
    "return_type": "xid",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gettransactionid()\n RETURNS xid\n LANGUAGE c\nAS '$libdir/postgis-3', $function$getTransactionID$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gidx_in",
    "arguments": "cstring",
    "return_type": "gidx",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gidx_in(cstring)\n RETURNS gidx\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gidx_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gidx_out",
    "arguments": "gidx",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gidx_out(gidx)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$gidx_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gserialized_gist_joinsel_2d",
    "arguments": "internal, oid, internal, smallint",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gserialized_gist_joinsel_2d(internal, oid, internal, smallint)\n RETURNS double precision\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_joinsel_2d$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "gserialized_gist_joinsel_nd",
    "arguments": "internal, oid, internal, smallint",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.gserialized_gist_joinsel_nd(internal, oid, internal, smallint)\n RETURNS double
precision\n LANGUAGE c\n PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$gserialized_gist_joinsel_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gserialized_gist_sel_2d",
    "arguments": "internal, oid, internal, integer",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gserialized_gist_sel_2d(internal,
oid, internal, integer)\n RETURNS double precision\n LANGUAGE c\n PARALLEL
SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_sel_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "gserialized_gist_sel_nd",
    "arguments": "internal, oid, internal, integer",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.gserialized_gist_sel_nd(internal,
oid, internal, integer)\n RETURNS double precision\n LANGUAGE c\n PARALLEL
SAFE\nAS '$libdir/postgis-3', $function$gserialized_gist_sel_nd$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec",
    "arguments": "halfvec, integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec(halfvec, integer,
boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_accum",
    "arguments": "double precision[], halfvec",
    "return_type": "double precision[]",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_accum(double
precision[], halfvec)\n RETURNS double precision[]\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_accum$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_add",
    "arguments": "halfvec, halfvec",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_add(halfvec, halfvec)\n
RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_add$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_avg",
    "arguments": "double precision[]",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_avg(double precision[])\n
RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_avg$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_cmp",
    "arguments": "halfvec, halfvec",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_cmp(halfvec, halfvec)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_cmp$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_combine",
    "arguments": "double precision[], double precision[]",
    "return_type": "double precision[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_combine(double
precision[], double precision[])\n RETURNS double precision[]\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$vector_combine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_concat",
```

```
    "arguments": "halfvec, halfvec",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_concat(halfvec,
halfvec)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec_concat$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_eq",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_eq(halfvec, halfvec)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_eq$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_ge",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_ge(halfvec, halfvec)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_ge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_gt",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_gt(halfvec, halfvec)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_gt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_in",
    "arguments": "cstring, oid, integer",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_in(cstring, oid, integer)\n
RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_in$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "halfvec_l2_squared_distance",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.halfvec_l2_squared_distance(halfvec, halfvec)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$halfvec_l2_squared_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_le",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_le(halfvec, halfvec)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_le$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_lt",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_lt(halfvec, halfvec)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_lt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_mul",
    "arguments": "halfvec, halfvec",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_mul(halfvec, halfvec)\n
RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_mul$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_ne",
    "arguments": "halfvec, halfvec",
    "return_type": "boolean",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_ne(halfvec, halfvec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_ne$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_negative_inner_product",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_negative_inner_product(halfvec, halfvec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_negative_inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_out",
    "arguments": "halfvec",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_out(halfvec)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_recv",
    "arguments": "internal, oid, integer",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_recv(internal, oid, integer)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_recv$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_send",
    "arguments": "halfvec",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_send(halfvec)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_send$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_spherical_distance",
```

```
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.halfvec_spherical_distance(halfvec, halfvec)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$halfvec_spherical_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_sub",
    "arguments": "halfvec, halfvec",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_sub(halfvec, halfvec)\n
RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_sub$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_to_float4",
    "arguments": "halfvec, integer, boolean",
    "return_type": "real[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_to_float4(halfvec, integer,
boolean)\n RETURNS real[]\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_to_float4$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_to_sparsevec",
    "arguments": "halfvec, integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_to_sparsevec(halfvec,
integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec_to_sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "halfvec_to_vector",
    "arguments": "halfvec, integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.halfvec_to_vector(halfvec,
integer, boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec_to_vector$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "halfvec_typmod_in",
   "arguments": "cstring[]",
   "return_type": "integer",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.halfvec_typmod_in(cstring[])\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_typmod_in$function$\n"
 },
 {
   "schema": "public",
   "function_name": "hamming_distance",
   "arguments": "bit, bit",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.hamming_distance(bit, bit)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$hamming_distance$function$\n"
 },
 {
   "schema": "public",
   "function_name": "hnsw_bit_support",
   "arguments": "internal",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.hnsw_bit_support(internal)\n
RETURNS internal\n LANGUAGE c\nAS '$libdir/vector',
$function$hnsw_bit_support$function$\n"
 },
 {
   "schema": "public",
   "function_name": "hnsw_halfvec_support",
   "arguments": "internal",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.hnsw_halfvec_support(internal)\n
RETURNS internal\n LANGUAGE c\nAS '$libdir/vector',
$function$hnsw_halfvec_support$function$\n"
 },
 {
   "schema": "public",
   "function_name": "hnsw_sparsevec_support",
   "arguments": "internal",
   "return_type": "internal",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION
public.hnsw_sparsevec_support(internal)\n RETURNS internal\n LANGUAGE c\nAS
'$libdir/vector', $function$hnsw_sparsevec_support$function$\n"
  },
  {
    "schema": "public",
    "function_name": "hnswhandler",
    "arguments": "internal",
    "return_type": "index_am_handler",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.hnswhandler(internal)\n
RETURNS index_am_handler\n LANGUAGE c\nAS '$libdir/vector',
$function$hnswhandler$function$\n"
  },
  {
    "schema": "public",
    "function_name": "inner_product",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.inner_product(halfvec, halfvec)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "inner_product",
    "arguments": "sparsevec, sparsevec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.inner_product(sparsevec,
sparsevec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "inner_product",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.inner_product(vector, vector)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "is_contained_2d",
    "arguments": "box2df, geometry",
```

```
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.is_contained_2d(box2df,
geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$gserialized_within_box2df_geom_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "is_contained_2d",
    "arguments": "geometry, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.is_contained_2d(geometry,
box2df)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 1\nAS $function$SELECT $2 OPERATOR(public.~) $1;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "is_contained_2d",
    "arguments": "box2df, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.is_contained_2d(box2df,
box2df)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3',
$function$gserialized_contains_box2df_box2df_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "ivfflat_bit_support",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.ivfflat_bit_support(internal)\n
RETURNS internal\n LANGUAGE c\nAS '$libdir/vector',
$function$ivfflat_bit_support$function$\n"
  },
  {
    "schema": "public",
    "function_name": "ivfflat_halfvec_support",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.ivfflat_halfvec_support(internal)\n
RETURNS internal\n LANGUAGE c\nAS '$libdir/vector',
$function$ivfflat_halfvec_support$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "ivfflathandler",
    "arguments": "internal",
    "return_type": "index_am_handler",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.ivfflathandler(internal)\n
RETURNS index_am_handler\n LANGUAGE c\nAS '$libdir/vector',
$function$ivfflathandler$function$\n"
  },
  {
    "schema": "public",
    "function_name": "jaccard_distance",
    "arguments": "bit, bit",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.jaccard_distance(bit, bit)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$jaccard_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "json",
    "arguments": "geometry",
    "return_type": "json",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.\"json\"(geometry)\n RETURNS
json\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$geometry_to_json$function$\n"
  },
  {
    "schema": "public",
    "function_name": "jsonb",
    "arguments": "geometry",
    "return_type": "jsonb",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.jsonb(geometry)\n RETURNS
jsonb\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$geometry_to_jsonb$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l1_distance",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l1_distance(halfvec, halfvec)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_l1_distance$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "l1_distance",
    "arguments": "sparsevec, sparsevec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l1_distance(sparsevec,
sparsevec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_l1_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l1_distance",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l1_distance(vector, vector)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$l1_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_distance",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_distance(vector, vector)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$l2_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_distance",
    "arguments": "halfvec, halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_distance(halfvec, halfvec)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_l2_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_distance",
    "arguments": "sparsevec, sparsevec",
    "return_type": "double precision",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.l2_distance(sparsevec,
sparsevec)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_l2_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_norm",
    "arguments": "sparsevec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_norm(sparsevec)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$sparsevec_l2_norm$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_norm",
    "arguments": "halfvec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_norm(halfvec)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$halfvec_l2_norm$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_normalize",
    "arguments": "sparsevec",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_normalize(sparsevec)\n
RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$sparsevec_l2_normalize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_normalize",
    "arguments": "vector",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_normalize(vector)\n
RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$l2_normalize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "l2_normalize",
    "arguments": "halfvec",
```

```
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.l2_normalize(halfvec)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_l2_normalize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "lockrow",
    "arguments": "text, text, text, text, timestamp without time zone",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.lockrow(text, text, text, text, timestamp without time zone)\n RETURNS integer\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tmyschema alias for $1;\n\tmytable alias for $2;\n\tmyrid   alias for $3;\n\tauthid alias for $4;\n\texpires alias for $5;\n\tret int;\n\tmytoid oid;\n\tmyrec RECORD;\n\nBEGIN\n\n\tIF NOT LongTransactionsEnabled() THEN\n\t\tRAISE EXCEPTION 'Long transaction support disabled, use EnableLongTransaction() to enable.';\n\tEND IF;\n\n\tEXECUTE 'DELETE FROM authorization_table WHERE expires < now()';\n\n\tSELECT c.oid INTO mytoid FROM pg_class c, pg_namespace n\n\t\tWHERE c.relname = mytable\n\t\tAND c.relnamespace = n.oid\n\t\tAND n.nspname = myschema;\n\n\t-- RAISE NOTICE 'toid: %', mytoid;\n\n\tFOR myrec IN SELECT * FROM authorization_table WHERE\n\t\ttoid = mytoid AND rid = myrid\n\tLOOP\n\t\tIF myrec.authid != authid THEN\n\t\t\tRETURN 0;\n\t\tELSE\n\t\t\tRETURN 1;\n\t\tEND IF;\n\tEND LOOP;\n\n\tEXECUTE 'INSERT INTO authorization_table VALUES ('||\n\t\tquote_literal(mytoid::text)||','||quote_literal(myrid)||\n\t\t','||quote_literal(expires::text)||\n\t\t','||quote_literal(authid) ||')';\n\n\tGET DIAGNOSTICS ret = ROW_COUNT;\n\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "lockrow",
    "arguments": "text, text, text, text",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.lockrow(text, text, text, text)\n RETURNS integer\n LANGUAGE sql\n STRICT\nAS $function$ SELECT LockRow($1, $2, $3, $4, now()::timestamp+'1:00'); $function$\n"
  },
  {
    "schema": "public",
    "function_name": "lockrow",
    "arguments": "text, text, text",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.lockrow(text, text, text)\n RETURNS integer\n LANGUAGE sql\n STRICT\nAS $function$ SELECT LockRow(current_schema(), $1, $2, $3, now()::timestamp+'1:00'); $function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "lockrow",
    "arguments": "text, text, text, timestamp without time zone",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.lockrow(text, text, text, timestamp without time zone)\n RETURNS integer\n LANGUAGE sql\n STRICT\nAS $function$ SELECT LockRow(current_schema(), $1, $2, $3, $4); $function$\n"
  },
  {
    "schema": "public",
    "function_name": "longtransactionsenabled",
    "arguments": "",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.longtransactionsenabled()\n RETURNS boolean\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\trec RECORD;\nBEGIN\n\tFOR rec IN SELECT oid FROM pg_class WHERE relname = 'authorized_tables'\n\tLOOP\n\t\treturn 't';\n\tEND LOOP;\n\treturn 'f';\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "match_products_v2",
    "arguments": "query_embedding vector, match_threshold double precision DEFAULT 0.75, match_count integer DEFAULT 5, filter_category_id integer DEFAULT NULL::integer",
    "return_type": "TABLE(id bigint, product_name character varying, product_description text, product_category_id integer, brand character varying, alternative_names jsonb, search_frequency integer, popularity_score numeric, similarity double precision)",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.match_products_v2(query_embedding vector, match_threshold double precision DEFAULT 0.75, match_count integer DEFAULT 5, filter_category_id integer DEFAULT NULL::integer)\n RETURNS TABLE(id bigint, product_name character varying, product_description text, product_category_id integer, brand character varying, alternative_names jsonb, search_frequency integer, popularity_score numeric, similarity double precision)\n LANGUAGE plpgsql\nAS $function$\nBEGIN\n  RETURN QUERY\n  SELECT\n    ml.id,\n    ml.product_name,\n    ml.product_description,\n    ml.product_category_id,\n    ml.brand,\n    ml.alternative_names,\n    ml.search_frequency,\n    ml.popularity_score,\n    1 - (ml.embedding_vector_v2 <=> query_embedding) AS similarity\n  FROM master_list ml\n  WHERE \n    ml.is_active = true\n    AND ml.embedding_vector_v2 IS NOT NULL\n    AND 1 - (ml.embedding_vector_v2 <=> query_embedding) > match_threshold\n    AND (filter_category_id IS NULL OR ml.product_category_id = filter_category_id)\n  ORDER BY ml.embedding_vector_v2 <=> query_embedding\n  LIMIT match_count;\nEND;\n$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "overlaps_2d",
    "arguments": "box2df, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_2d(box2df, box2df)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$gserialized_contains_box2df_box2df_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_2d",
    "arguments": "geometry, box2df",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_2d(geometry, box2df)\n
RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
1\nAS $function$SELECT $2 OPERATOR(public.&&) $1;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_2d",
    "arguments": "box2df, geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_2d(box2df, geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$gserialized_overlaps_box2df_geom_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_geog",
    "arguments": "geography, gidx",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_geog(geography,
gidx)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE STRICT\nAS
$function$SELECT $2 OPERATOR(public.&&) $1;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_geog",
    "arguments": "gidx, gidx",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_geog(gidx, gidx)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE STRICT\nAS '$libdir/postgis-3',
$function$gserialized_gidx_gidx_overlaps$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "overlaps_geog",
    "arguments": "gidx, geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_geog(gidx,
geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE STRICT\nAS
'$libdir/postgis-3', $function$gserialized_gidx_geog_overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_nd",
    "arguments": "geometry, gidx",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_nd(geometry, gidx)\n
RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
1\nAS $function$SELECT $2 OPERATOR(public.&&&) $1;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_nd",
    "arguments": "gidx, geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_nd(gidx, geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$gserialized_gidx_geom_overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "overlaps_nd",
    "arguments": "gidx, gidx",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.overlaps_nd(gidx, gidx)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$gserialized_gidx_gidx_overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "path",
    "arguments": "geometry",
    "return_type": "path",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.path(geometry)\n RETURNS path\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$geometry_to_path$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asflatgeobuf_finalfn",
    "arguments": "internal",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asflatgeobuf_finalfn(internal)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asflatgeobuf_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asflatgeobuf_transfn",
    "arguments": "internal, anyelement",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asflatgeobuf_transfn(internal, anyelement)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$pgis_asflatgeobuf_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asflatgeobuf_transfn",
    "arguments": "internal, anyelement, boolean, text",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asflatgeobuf_transfn(internal, anyelement, boolean, text)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$pgis_asflatgeobuf_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asflatgeobuf_transfn",
    "arguments": "internal, anyelement, boolean",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asflatgeobuf_transfn(internal, anyelement, boolean)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$pgis_asflatgeobuf_transfn$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "pgis_asgeobuf_finalfn",
   "arguments": "internal",
   "return_type": "bytea",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.pgis_asgeobuf_finalfn(internal)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asgeobuf_finalfn$function$\n"
 },
 {
   "schema": "public",
   "function_name": "pgis_asgeobuf_transfn",
   "arguments": "internal, anyelement",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.pgis_asgeobuf_transfn(internal, anyelement)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$pgis_asgeobuf_transfn$function$\n"
 },
 {
   "schema": "public",
   "function_name": "pgis_asgeobuf_transfn",
   "arguments": "internal, anyelement, text",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.pgis_asgeobuf_transfn(internal, anyelement, text)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$pgis_asgeobuf_transfn$function$\n"
 },
 {
   "schema": "public",
   "function_name": "pgis_asmvt_combinefn",
   "arguments": "internal, internal",
   "return_type": "internal",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_combinefn(internal, internal)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_combinefn$function$\n"
 },
 {
   "schema": "public",
   "function_name": "pgis_asmvt_deserialfn",
   "arguments": "bytea, internal",
   "return_type": "internal",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_deserialfn(bytea, internal)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_deserialfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_finalfn",
    "arguments": "internal",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_finalfn(internal)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_serialfn",
    "arguments": "internal",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_serialfn(internal)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_serialfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_transfn",
    "arguments": "internal, anyelement, text",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_transfn(internal, anyelement, text)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_transfn",
    "arguments": "internal, anyelement, text, integer, text",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_transfn(internal, anyelement, text, integer, text)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_transfn",
```

```
    "arguments": "internal, anyelement",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_transfn(internal,
anyelement)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST
500\nAS '$libdir/postgis-3', $function$pgis_asmvt_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_transfn",
    "arguments": "internal, anyelement, text, integer",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_transfn(internal,
anyelement, text, integer)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_asmvt_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_asmvt_transfn",
    "arguments": "internal, anyelement, text, integer, text, text",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.pgis_asmvt_transfn(internal,
anyelement, text, integer, text, text)\n RETURNS internal\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_asmvt_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_accum_transfn",
    "arguments": "internal, geometry",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_accum_transfn(internal, geometry)\n RETURNS internal\n
LANGUAGE c\n PARALLEL SAFE COST 50\nAS '$libdir/postgis-3',
$function$pgis_geometry_accum_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_accum_transfn",
    "arguments": "internal, geometry, double precision",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_accum_transfn(internal, geometry, double precision)\n RETURNS
```

internal\n LANGUAGE c\n PARALLEL SAFE COST 50\nAS '$libdir/postgis-3',
$function$pgis_geometry_accum_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_accum_transfn",
    "arguments": "internal, geometry, double precision, integer",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_accum_transfn(internal, geometry, double precision, integer)\n
RETURNS internal\n LANGUAGE c\n PARALLEL SAFE COST 50\nAS '$libdir/postgis-3',
$function$pgis_geometry_accum_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_clusterintersecting_finalfn",
    "arguments": "internal",
    "return_type": "geometry[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_clusterintersecting_finalfn(internal)\n RETURNS geometry[]\n
LANGUAGE c\n PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_geometry_clusterintersecting_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_clusterwithin_finalfn",
    "arguments": "internal",
    "return_type": "geometry[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_clusterwithin_finalfn(internal)\n RETURNS geometry[]\n LANGUAGE
c\n PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_geometry_clusterwithin_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_collect_finalfn",
    "arguments": "internal",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_collect_finalfn(internal)\n RETURNS geometry\n LANGUAGE c\n
PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_geometry_collect_finalfn$function$\n"
  },
  {

```
    "schema": "public",
    "function_name": "pgis_geometry_makeline_finalfn",
    "arguments": "internal",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_makeline_finalfn(internal)\n RETURNS geometry\n LANGUAGE c\n
PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_geometry_makeline_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_polygonize_finalfn",
    "arguments": "internal",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_polygonize_finalfn(internal)\n RETURNS geometry\n LANGUAGE c\n
PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$pgis_geometry_polygonize_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_combinefn",
    "arguments": "internal, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_combinefn(internal, internal)\n RETURNS internal\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$pgis_geometry_union_parallel_combinefn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_deserialfn",
    "arguments": "bytea, internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_deserialfn(bytea, internal)\n RETURNS internal\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$pgis_geometry_union_parallel_deserialfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_finalfn",
    "arguments": "internal",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_finalfn(internal)\n RETURNS geometry\n LANGUAGE
c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$pgis_geometry_union_parallel_finalfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_serialfn",
    "arguments": "internal",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_serialfn(internal)\n RETURNS bytea\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$pgis_geometry_union_parallel_serialfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_transfn",
    "arguments": "internal, geometry",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_transfn(internal, geometry)\n RETURNS internal\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE\nAS '$libdir/postgis-3',
$function$pgis_geometry_union_parallel_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "pgis_geometry_union_parallel_transfn",
    "arguments": "internal, geometry, double precision",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.pgis_geometry_union_parallel_transfn(internal, geometry, double precision)\n
RETURNS internal\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS
'$libdir/postgis-3', $function$pgis_geometry_union_parallel_transfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "point",
    "arguments": "geometry",
    "return_type": "point",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.point(geometry)\n RETURNS
point\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$geometry_to_point$function$\n"
```

```
 },
 {
   "schema": "public",
   "function_name": "polygon",
   "arguments": "geometry",
   "return_type": "polygon",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.polygon(geometry)\n RETURNS
polygon\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$geometry_to_polygon$function$\n"
 },
 {
   "schema": "public",
   "function_name": "populate_geometry_columns",
   "arguments": "tbl_oid oid, use_typmod boolean DEFAULT true",
   "return_type": "integer",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.populate_geometry_columns(tbl_oid oid, use_typmod boolean DEFAULT true)\n
RETURNS integer\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\tgcs\t\t
RECORD;\n\tgc\t\t  RECORD;\n\tgc_old\t  RECORD;\n\tgsrid\t   integer;\n\tgndims\t
integer;\n\tgtype\t   text;\n\tquery\t   text;\n\tgc_is_valid
boolean;\n\tinserted\tinteger;\n\tconstraint_successful boolean := false;\n\nBEGIN\n\tinserted
:= 0;\n\n\t-- Iterate through all geometry columns in this table\n\tFOR gcs IN\n\tSELECT
n.nspname, c.relname, a.attname, c.relkind\n\t\tFROM pg_class c,\n\t\t\t pg_attribute a,\n\t\t\t
pg_type t,\n\t\t\t pg_namespace n\n\t\tWHERE c.relkind IN('r', 'f', 'p')\n\t\tAND t.typname =
'geometry'\n\t\tAND a.attisdropped = false\n\t\tAND a.atttypid = t.oid\n\t\tAND a.attrelid =
c.oid\n\t\tAND c.relnamespace = n.oid\n\t\tAND n.nspname NOT ILIKE 'pg_temp%'\n\t\tAND
c.oid = tbl_oid\n\tLOOP\n\n\t\tRAISE DEBUG 'Processing column %.%.%', gcs.nspname,
gcs.relname, gcs.attname;\n\n\t\tgc_is_valid := true;\n\t\t-- Find the srid, coord_dimension,
and type of current geometry\n\t\t-- in geometry_columns -- which is now a
view\n\n\t\tSELECT type, srid, coord_dimension, gcs.relkind INTO gc_old\n\t\t\tFROM
geometry_columns\n\t\t\tWHERE f_table_schema = gcs.nspname AND f_table_name =
gcs.relname AND f_geometry_column = gcs.attname;\n\n\t\tIF upper(gc_old.type) =
'GEOMETRY' THEN\n\t\t-- This is an unconstrained geometry we need to do
something\n\t\t-- We need to figure out what to set the type by inspecting the
data\n\t\t\tEXECUTE 'SELECT public.ST_srid(' || quote_ident(gcs.attname) || ') As srid,
public.GeometryType(' || quote_ident(gcs.attname) || ') As type, public.ST_NDims(' ||
quote_ident(gcs.attname) || ') As dims ' ||\n\t\t\t\t\t ' FROM ONLY ' ||
quote_ident(gcs.nspname) || '.' || quote_ident(gcs.relname) ||\n\t\t\t\t\t ' WHERE ' ||
quote_ident(gcs.attname) || ' IS NOT NULL LIMIT 1;'\n\t\t\t\tINTO gc;\n\t\t\tIF gc IS NULL
THEN -- there is no data so we can not determine geometry type\n\t\t\t\tRAISE WARNING
'No data in table %.%, so no information to determine geometry type and srid', gcs.nspname,
gcs.relname;\n\t\t\t\tRETURN 0;\n\t\t\tEND IF;\n\t\t\tgsrid := gc.srid; gtype := gc.type; gndims
:= gc.dims;\n\n\t\t\tIF use_typmod THEN\n\t\t\t\tBEGIN\n\t\t\t\t\tEXECUTE 'ALTER TABLE ' ||
quote_ident(gcs.nspname) || '.' || quote_ident(gcs.relname) || ' ALTER COLUMN ' ||
quote_ident(gcs.attname) ||\n\t\t\t\t\t' TYPE geometry(' || postgis_type_name(gtype, gndims,
true) || ', ' || gsrid::text  || ') ';\n\t\t\t\tinserted := inserted +
```

```
1;\n\t\t\t\tEXCEPTION\n\t\t\t\t\t\tWHEN invalid_parameter_value OR feature_not_supported
THEN\n\t\t\t\t\t\tRAISE WARNING 'Could not convert "%" in "%.%" to use typmod with srid
%, type %: %', quote_ident(gcs.attname), quote_ident(gcs.nspname),
quote_ident(gcs.relname), gsrid, postgis_type_name(gtype, gndims, true),
SQLERRM;\n\t\t\t\t\t\tgc_is_valid := false;\n\t\t\t\tEND;\n\n\t\tELSE\n\t\t\t-- Try to apply
srid check to column\n\t\t\tconstraint_successful = false;\n\t\t\tIF (gsrid > 0 AND
postgis_constraint_srid(gcs.nspname, gcs.relname,gcs.attname) IS NULL )
THEN\n\t\t\t\tBEGIN\n\t\t\t\t\tEXECUTE 'ALTER TABLE ONLY ' ||
quote_ident(gcs.nspname) || '.' || quote_ident(gcs.relname) ||\n\t\t\t\t\t ' ADD
CONSTRAINT ' || quote_ident('enforce_srid_' || gcs.attname) ||\n\t\t\t\t\t ' CHECK
(ST_srid(' || quote_ident(gcs.attname) || ') = ' || gsrid || ')';\n\t\t\t\tconstraint_successful :=
true;\n\t\t\t\tEXCEPTION\n\t\t\t\t\tWHEN check_violation THEN\n\t\t\t\t\tRAISE
WARNING 'Not inserting "%" in "%.%" into geometry_columns: could not apply constraint
CHECK (st_srid(%) = %)', quote_ident(gcs.attname), quote_ident(gcs.nspname),
quote_ident(gcs.relname), quote_ident(gcs.attname), gsrid;\n\t\t\t\t\tgc_is_valid :=
false;\n\t\t\t\tEND;\n\t\t\tEND IF;\n\n\t\t-- Try to apply ndims check to column\n\t\tIF
(gndims IS NOT NULL AND postgis_constraint_dims(gcs.nspname,
gcs.relname,gcs.attname) IS NULL ) THEN\n\t\t\tBEGIN\n\t\t\t\tEXECUTE 'ALTER
TABLE ONLY ' || quote_ident(gcs.nspname) || '.' || quote_ident(gcs.relname) || '\n\t\t\t\t ADD CONSTRAINT ' || quote_ident('enforce_dims_' || gcs.attname) || '\n\t\t\t\t CHECK
(st_ndims(' || quote_ident(gcs.attname) || ') = '||gndims||')';\n\t\t\t\tconstraint_successful :=
true;\n\t\t\t\tEXCEPTION\n\t\t\t\t\tWHEN check_violation THEN\n\t\t\t\t\tRAISE
WARNING 'Not inserting "%" in "%.%" into geometry_columns: could not apply constraint
CHECK (st_ndims(%) = %)', quote_ident(gcs.attname), quote_ident(gcs.nspname),
quote_ident(gcs.relname), quote_ident(gcs.attname), gndims;\n\t\t\t\t\tgc_is_valid :=
false;\n\t\t\t\tEND;\n\t\t\tEND IF;\n\n\t\t-- Try to apply geometrytype check to
column\n\t\tIF (gtype IS NOT NULL AND postgis_constraint_type(gcs.nspname,
gcs.relname,gcs.attname) IS NULL ) THEN\n\t\t\tBEGIN\n\t\t\t\tEXECUTE 'ALTER
TABLE ONLY ' || quote_ident(gcs.nspname) || '.' || quote_ident(gcs.relname) ||
'\n\t\t\t\tADD CONSTRAINT ' || quote_ident('enforce_geotype_' || gcs.attname) ||
'\n\t\t\t\tCHECK (geometrytype(' || quote_ident(gcs.attname) || ') = ' || quote_literal(gtype) ||
')';\n\t\t\t\tconstraint_successful := true;\n\t\t\tEXCEPTION\n\t\t\t\tWHEN
check_violation THEN\n\t\t\t\t-- No geometry check can be applied. This column contains
a number of geometry types.\n\t\t\t\tRAISE WARNING 'Could not add geometry type
check (%) to table column: %.%.%', gtype,
quote_ident(gcs.nspname),quote_ident(gcs.relname),quote_ident(gcs.attname);\n\t\t\tEN
D;\n\t\tEND IF;\n\t\t\t --only count if we were successful in applying at least one
constraint\n\t\tIF constraint_successful THEN\n\t\t\tinserted := inserted + 1;\n\t\tEND
IF;\n\tEND IF;\n\tEND IF;\n\nEND LOOP;\n\nRETURN
inserted;\nEND\n\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "populate_geometry_columns",
    "arguments": "use_typmod boolean DEFAULT true",
    "return_type": "text",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION
public.populate_geometry_columns(use_typmod boolean DEFAULT true)\n RETURNS
text\n LANGUAGE plpgsql\nAS
$function$\nDECLARE\n\tinserted\tinteger;\n\toldcount\tinteger;\n\tprobed\t
integer;\n\tstale\t   integer;\n\tgcs\t\t RECORD;\n\tgc\t\t  RECORD;\n\tgsrid\t
integer;\n\tgndims\t  integer;\n\tgtype\t   text;\n\tquery\t   text;\n\tgc_is_valid
boolean;\n\nBEGIN\n\tSELECT count(*) INTO oldcount FROM
public.geometry_columns;\n\tinserted := 0;\n\n\t-- Count the number of geometry columns in
all tables and views\n\tSELECT count(DISTINCT c.oid) INTO probed\n\tFROM pg_class
c,\n\t\t pg_attribute a,\n\t\t pg_type t,\n\t\t pg_namespace n\n\tWHERE c.relkind IN('r','v','f',
'p')\n\t\tAND t.typname = 'geometry'\n\t\tAND a.attisdropped = false\n\t\tAND a.atttypid =
t.oid\n\t\tAND a.attrelid = c.oid\n\t\tAND c.relnamespace = n.oid\n\t\tAND n.nspname NOT
ILIKE 'pg_temp%' AND c.relname != 'raster_columns' ;\n\n\t-- Iterate through all non-dropped
geometry columns\n\tRAISE DEBUG 'Processing Tables.....';\n\n\tFOR gcs IN\n\tSELECT
DISTINCT ON (c.oid) c.oid, n.nspname, c.relname\n\t\tFROM pg_class c,\n\t\t\t pg_attribute
a,\n\t\t\t pg_type t,\n\t\t\t pg_namespace n\n\t\tWHERE c.relkind IN( 'r', 'f', 'p')\n\t\tAND
t.typname = 'geometry'\n\t\tAND a.attisdropped = false\n\t\tAND a.atttypid = t.oid\n\t\tAND
a.attrelid = c.oid\n\t\tAND c.relnamespace = n.oid\n\t\tAND n.nspname NOT ILIKE
'pg_temp%' AND c.relname != 'raster_columns'\n\tLOOP\n\n\t\tinserted := inserted +
public.populate_geometry_columns(gcs.oid, use_typmod);\n\tEND LOOP;\n\n\tIF oldcount >
inserted THEN\n\t\tstale = oldcount-inserted;\n\tELSE\n\t\tstale = 0;\n\tEND IF;\n\n\tRETURN
'probed:' ||probed|| ' inserted:'||inserted;\nEND\n\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_addbbox",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_addbbox(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_addBBOX$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_cache_bbox",
    "arguments": "",
    "return_type": "trigger",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_cache_bbox()\n
RETURNS trigger\n LANGUAGE c\nAS '$libdir/postgis-3',
$function$cache_bbox$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_constraint_dims",
    "arguments": "geomschema text, geomtable text, geomcolumn text",
    "return_type": "integer",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.postgis_constraint_dims(geomschema text, geomtable text, geomcolumn text)\n
RETURNS integer\n LANGUAGE sql\n STABLE PARALLEL SAFE STRICT COST 500\nAS
$function$\nSELECT  replace(split_part(s.consrc, ' = ', 2), ')', '')::integer\n\t\t FROM pg_class
c, pg_namespace n, pg_attribute a\n\t\t , (SELECT connamespace, conrelid, conkey,
pg_get_constraintdef(oid) As consrc\n\t\t\tFROM pg_constraint) AS s\n\t\t WHERE
n.nspname = $1\n\t\t AND c.relname = $2\n\t\t AND a.attname = $3\n\t\t AND a.attrelid =
c.oid\n\t\t AND s.connamespace = n.oid\n\t\t AND s.conrelid = c.oid\n\t\t AND a.attnum =
ANY (s.conkey)\n\t\t AND s.consrc LIKE '%ndims(% = %';\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_constraint_srid",
    "arguments": "geomschema text, geomtable text, geomcolumn text",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.postgis_constraint_srid(geomschema text, geomtable text, geomcolumn text)\n
RETURNS integer\n LANGUAGE sql\n STABLE PARALLEL SAFE STRICT COST 500\nAS
$function$\nSELECT replace(replace(split_part(s.consrc, ' = ', 2), ')', ''), '(', '')::integer\n\t\t
FROM pg_class c, pg_namespace n, pg_attribute a\n\t\t , (SELECT connamespace,
conrelid, conkey, pg_get_constraintdef(oid) As consrc\n\t\t\tFROM pg_constraint) AS s\n\t\t
WHERE n.nspname = $1\n\t\t AND c.relname = $2\n\t\t AND a.attname = $3\n\t\t AND
a.attrelid = c.oid\n\t\t AND s.connamespace = n.oid\n\t\t AND s.conrelid = c.oid\n\t\t AND
a.attnum = ANY (s.conkey)\n\t\t AND s.consrc LIKE '%srid(% = %';\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_constraint_type",
    "arguments": "geomschema text, geomtable text, geomcolumn text",
    "return_type": "character varying",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.postgis_constraint_type(geomschema text, geomtable text, geomcolumn text)\n
RETURNS character varying\n LANGUAGE sql\n STABLE PARALLEL SAFE STRICT COST
500\nAS $function$\nSELECT  replace(split_part(s.consrc, '''', 2), ')', '')::varchar\n\t\t FROM
pg_class c, pg_namespace n, pg_attribute a\n\t\t , (SELECT connamespace, conrelid,
conkey, pg_get_constraintdef(oid) As consrc\n\t\t\tFROM pg_constraint) AS s\n\t\t WHERE
n.nspname = $1\n\t\t AND c.relname = $2\n\t\t AND a.attname = $3\n\t\t AND a.attrelid =
c.oid\n\t\t AND s.connamespace = n.oid\n\t\t AND s.conrelid = c.oid\n\t\t AND a.attnum =
ANY (s.conkey)\n\t\t AND s.consrc LIKE '%geometrytype(% = %';\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_dropbbox",
    "arguments": "geometry",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_dropbbox(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_dropBBOX$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_extensions_upgrade",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_extensions_upgrade()\n
RETURNS text\n LANGUAGE plpgsql\nAS $function$\nDECLARE\n\trec record;\n\tsql
text;\n\tvar_schema text;\n\ttarget_version text; -- TODO: optionally take as
argument\nBEGIN\n\n\tFOR rec IN\n\t\tSELECT name, default_version,
installed_version\n\t\tFROM pg_catalog.pg_available_extensions\n\t\tWHERE name IN
(\n\t\t\t'postgis',\n\t\t\t'postgis_raster',\n\t\t\t'postgis_sfcgal',\n\t\t\t'postgis_topology',\n\t\t\t'post
gis_tiger_geocoder'\n\t\t)\n\t\tORDER BY length(name) -- this is to make sure 'postgis' is first
!\n\tLOOP --{\n\n\t\tIF target_version IS NULL THEN\n\t\t\ttarget_version :=
rec.default_version;\n\t\tEND IF;\n\n\t\tIF rec.installed_version IS NULL THEN --{\n\t\t\t-- If
the support installed by available extension\n\t\t\t-- is found unpackaged, we package
it\n\t\t\tIF --{\n\t\t\t\t -- PostGIS is always available (this function is part of it)\n\t\t\t\t rec.name
= 'postgis'\n\n\t\t\t\t -- PostGIS raster is available if type 'raster' exists\n\t\t\t\t OR ( rec.name
= 'postgis_raster' AND EXISTS (\n\t\t\t\t\t\t\tSELECT 1 FROM
pg_catalog.pg_type\n\t\t\t\t\t\t\tWHERE typname = 'raster' ) )\n\n\t\t\t\t -- PostGIS SFCGAL is
availble if\n\t\t\t\t -- 'postgis_sfcgal_version' function exists\n\t\t\t\t OR ( rec.name =
'postgis_sfcgal' AND EXISTS (\n\t\t\t\t\t\t\tSELECT 1 FROM
pg_catalog.pg_proc\n\t\t\t\t\t\t\tWHERE proname = 'postgis_sfcgal_version' ) )\n\n\t\t\t\t --
PostGIS Topology is available if\n\t\t\t\t -- 'topology.topology' table exists\n\t\t\t\t -- NOTE:
watch out for https://trac.osgeo.org/postgis/ticket/2503\n\t\t\t\t OR ( rec.name =
'postgis_topology' AND EXISTS (\n\t\t\t\t\t\t\tSELECT 1 FROM pg_catalog.pg_class
c\n\t\t\t\t\t\t\tJOIN pg_catalog.pg_namespace n ON (c.relnamespace = n.oid
)\n\t\t\t\t\t\t\tWHERE n.nspname = 'topology' AND c.relname = 'topology') )\n\n\t\t\t\t OR (
rec.name = 'postgis_tiger_geocoder' AND EXISTS (\n\t\t\t\t\t\t\tSELECT 1 FROM
pg_catalog.pg_class c\n\t\t\t\t\t\t\tJOIN pg_catalog.pg_namespace n ON (c.relnamespace =
n.oid )\n\t\t\t\t\t\t\tWHERE n.nspname = 'tiger' AND c.relname = 'geocode_settings')
)\n\t\t\tTHEN --}{\n\t\t\t-- Force install in same schema as postgis\n\t\t\t\tSELECT INTO
var_schema n.nspname\n\t\t\t\t  FROM pg_namespace n, pg_proc p\n\t\t\t\t  WHERE
p.proname = 'postgis_full_version'\n\t\t\t\tAND n.oid = p.pronamespace\n\t\t\t  LIMIT
1;\n\t\t\tIF rec.name NOT IN('postgis_topology',
'postgis_tiger_geocoder')\n\t\t\tTHEN\n\t\t\t\tsql := format(\n\t\t\t\t  'CREATE
EXTENSION %1$I SCHEMA %2$I VERSION unpackaged;'\n\t\t\t\t  'ALTER
EXTENSION %1$I UPDATE TO %3$I',\n\t\t\t\t  rec.name, var_schema,
target_version);\n\t\t\tELSE\n\t\t\t\tsql := format(\n\t\t\t\t 'CREATE EXTENSION %1$I
VERSION unpackaged;'\n\t\t\t\t 'ALTER EXTENSION %1$I UPDATE TO
%2$I',\n\t\t\t\t rec.name, target_version);\n\t\t\tEND IF;\n\t\t\tRAISE NOTICE
'Packaging and updating %', rec.name;\n\t\t\tRAISE DEBUG '%', sql;\n\t\t\tEXECUTE
sql;\n\t\tELSE\n\t\t\tRAISE DEBUG 'Skipping % (not in use)', rec.name;\n\t\tEND
```

IF;\n\t\tELSE -- IF target_version != rec.installed_version THEN --}{\n\t\t\tsql = '';\n\t\t\t-- If logged in as super user\n\t\t\t-- force an update regardless if at target version, no downgrade allowed\n\t\t\tIF (SELECT usesuper FROM pg_user WHERE usename = CURRENT_USER)\n\t\t\t\t\t\tAND pg_catalog.substring(target_version, '[0-9]+\\.[0-9]+\\.[0-9]+')\n\t\t\t\t\t\t>= pg_catalog.substring(rec.installed_version, '[0-9]+\\.[0-9]+\\.[0-9]+')\n\t\t\tTHEN\n\t\t\t\tsql = format(\n\t\t\t\t\t'UPDATE pg_catalog.pg_extension SET extversion = ''ANY'' WHERE extname = %1$L;'\n\t\t\t\t\t'ALTER EXTENSION %1$I UPDATE TO %2$I',\n\t\t\t\t\trec.name, target_version\n\t\t\t\t);\n\t\t\t-- sandboxed users do standard upgrade\n\t\t\tELSE\n\t\t\t\tsql = format(\n\t\t\t\t\t'ALTER EXTENSION %1$I UPDATE TO %2$I',\n\t\t\t\t\trec.name, target_version\n\t\t\t\t);\n\t\t\tEND IF;\n\t\t\tRAISE NOTICE 'Updating extension %%',\n\t\t\t\trec.name, rec.installed_version;\n\t\t\tRAISE DEBUG '%', sql;\n\t\t\tEXECUTE sql;\n\t\tEND IF; --}\n\n\tEND LOOP; --}\n\n\tRETURN format(\n\t\t'Upgrade to version %s completed, run SELECT postgis_full_version(); for details',\n\t\t\ttarget_version\n\t);\n\n\nEND\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_full_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_full_version()\n RETURNS text\n LANGUAGE plpgsql\n IMMUTABLE\nAS $function$\nDECLARE\n\tlibver text;\n\tlibrev text;\n\tprojver text;\n\tgeosver text;\n\tsfcgalver text;\n\tgdalver text := NULL;\n\tlibxmlver text;\n\tliblwgeomver text;\n\tdbproc text;\n\trelproc text;\n\tfullver text;\n\trast_lib_ver text := NULL;\n\trast_scr_ver text := NULL;\n\ttopo_scr_ver text := NULL;\n\tjson_lib_ver text;\n\tprotobuf_lib_ver text;\n\twagyu_lib_ver text;\n\tsfcgal_lib_ver text;\n\tsfcgal_scr_ver text;\n\tpgsql_scr_ver text;\n\tpgsql_ver text;\n\tcore_is_extension bool;\nBEGIN\n\tSELECT public.postgis_lib_version() INTO libver;\n\tSELECT public.postgis_proj_version() INTO projver;\n\tSELECT public.postgis_geos_version() INTO geosver;\n\tSELECT public.postgis_libjson_version() INTO json_lib_ver;\n\tSELECT public.postgis_libprotobuf_version() INTO protobuf_lib_ver;\n\tSELECT public.postgis_wagyu_version() INTO wagyu_lib_ver;\n\tSELECT public._postgis_scripts_pgsql_version() INTO pgsql_scr_ver;\n\tSELECT public._postgis_pgsql_version() INTO pgsql_ver;\n\tBEGIN\n\t\tSELECT public.postgis_gdal_version() INTO gdalver;\n\tEXCEPTION\n\t\tWHEN undefined_function THEN\n\t\t\tRAISE DEBUG 'Function postgis_gdal_version() not found. Is raster support enabled and rtpostgis.sql installed?';\n\tEND;\n\tBEGIN\n\t\tSELECT public.postgis_sfcgal_full_version() INTO sfcgalver;\n\t\tBEGIN\n\t\t\tSELECT public.postgis_sfcgal_scripts_installed() INTO sfcgal_scr_ver;\n\t\tEXCEPTION\n\t\t\tWHEN undefined_function THEN\n\t\t\t\tsfcgal_scr_ver := 'missing';\n\t\tEND;\n\tEXCEPTION\n\t\tWHEN undefined_function THEN\n\t\t\tRAISE DEBUG 'Function postgis_sfcgal_scripts_installed() not found. Is sfcgal support enabled and sfcgal.sql installed?';\n\tEND;\n\tSELECT public.postgis_liblwgeom_version() INTO liblwgeomver;\n\tSELECT public.postgis_libxml_version() INTO libxmlver;\n\tSELECT public.postgis_scripts_installed() INTO dbproc;\n\tSELECT public.postgis_scripts_released() INTO relproc;\n\tSELECT public.postgis_lib_revision() INTO librev;\n\tBEGIN\n\t\tSELECT

topology.postgis_topology_scripts_installed() INTO topo_scr_ver;\n\tEXCEPTION\n\t\tWHEN undefined_function OR invalid_schema_name THEN\n\t\t\tRAISE DEBUG 'Function postgis_topology_scripts_installed() not found. Is topology support enabled and topology.sql installed?';\n\t\tWHEN insufficient_privilege THEN\n\t\t\tRAISE NOTICE 'Topology support cannot be inspected. Is current user granted USAGE on schema \"topology\" ?';\n\t\tWHEN OTHERS THEN\n\t\t\tRAISE NOTICE 'Function postgis_topology_scripts_installed() could not be called: % (%)', SQLERRM, SQLSTATE;\n\tEND;\n\n\tBEGIN\n\t\tSELECT postgis_raster_scripts_installed() INTO rast_scr_ver;\n\tEXCEPTION\n\t\tWHEN undefined_function THEN\n\t\t\tRAISE DEBUG 'Function postgis_raster_scripts_installed() not found. Is raster support enabled and rtpostgis.sql installed?';\n\t\tWHEN OTHERS THEN\n\t\t\tRAISE NOTICE 'Function postgis_raster_scripts_installed() could not be called: % (%)', SQLERRM, SQLSTATE;\n\tEND;\n\n\tBEGIN\n\t\tSELECT public.postgis_raster_lib_version() INTO rast_lib_ver;\n\tEXCEPTION\n\t\tWHEN undefined_function THEN\n\t\t\tRAISE DEBUG 'Function postgis_raster_lib_version() not found. Is raster support enabled and rtpostgis.sql installed?';\n\t\tWHEN OTHERS THEN\n\t\t\tRAISE NOTICE 'Function postgis_raster_lib_version() could not be called: % (%)', SQLERRM, SQLSTATE;\n\tEND;\n\n\tfullver = 'POSTGIS=\"' || libver;\n\n\tIF librev IS NOT NULL THEN\n\t\tfullver = fullver || ' ' || librev;\n\tEND IF;\n\n\tfullver = fullver || '\"';\n\n\tIF EXISTS (\n\t\tSELECT * FROM pg_catalog.pg_extension\n\t\tWHERE extname = 'postgis')\n\tTHEN\n\t\t\tfullver = fullver || ' [EXTENSION]';\n\t\t\tcore_is_extension := true;\n\tELSE\n\t\t\tcore_is_extension := false;\n\tEND IF;\n\n\tIF liblwgeomver != relproc THEN\n\t\tfullver = fullver || ' (liblwgeom version mismatch: \"' || liblwgeomver || '\")';\n\tEND IF;\n\n\tfullver = fullver || ' PGSQL=\"' || pgsql_scr_ver || '\"';\n\tIF pgsql_scr_ver != pgsql_ver THEN\n\t\tfullver = fullver || ' (procs need upgrade for use with PostgreSQL \"' || pgsql_ver || '\")';\n\tEND IF;\n\n\tIF geosver IS NOT NULL THEN\n\t\tfullver = fullver || ' GEOS=\"' || geosver || '\"';\n\tEND IF;\n\n\tIF sfcgalver IS NOT NULL THEN\n\t\tfullver = fullver || ' SFCGAL=\"' || sfcgalver || '\"';\n\tEND IF;\n\n\tIF projver IS NOT NULL THEN\n\t\tfullver = fullver || ' PROJ=\"' || projver || '\"';\n\tEND IF;\n\n\tIF gdalver IS NOT NULL THEN\n\t\tfullver = fullver || ' GDAL=\"' || gdalver || '\"';\n\tEND IF;\n\n\tIF libxmlver IS NOT NULL THEN\n\t\tfullver = fullver || ' LIBXML=\"' || libxmlver || '\"';\n\tEND IF;\n\n\tIF json_lib_ver IS NOT NULL THEN\n\t\tfullver = fullver || ' LIBJSON=\"' || json_lib_ver || '\"';\n\tEND IF;\n\n\tIF protobuf_lib_ver IS NOT NULL THEN\n\t\tfullver = fullver || ' LIBPROTOBUF=\"' || protobuf_lib_ver || '\"';\n\tEND IF;\n\n\tIF wagyu_lib_ver IS NOT NULL THEN\n\t\tfullver = fullver || ' WAGYU=\"' || wagyu_lib_ver || '\"';\n\tEND IF;\n\n\tIF dbproc != relproc THEN\n\t\tfullver = fullver || ' (core procs from \"' || dbproc || '\" need upgrade)';\n\tEND IF;\n\n\tIF topo_scr_ver IS NOT NULL THEN\n\t\tfullver = fullver || ' TOPOLOGY';\n\t\tIF topo_scr_ver != relproc THEN\n\t\t\tfullver = fullver || ' (topology procs from \"' || topo_scr_ver || '\" need upgrade)';\n\t\tEND IF;\n\t\tIF core_is_extension AND NOT EXISTS (\n\t\t\tSELECT * FROM pg_catalog.pg_extension\n\t\t\tWHERE extname = 'postgis_topology')\n\t\tTHEN\n\t\t\t\tfullver = fullver || ' [UNPACKAGED!]';\n\t\tEND IF;\n\tEND IF;\n\n\tIF rast_lib_ver IS NOT NULL THEN\n\t\tfullver = fullver || ' RASTER';\n\tIF rast_lib_ver != relproc THEN\n\t\t\tfullver = fullver || ' (raster lib from \"' || rast_lib_ver || '\" need upgrade)';\n\t\tEND IF;\n\tIF core_is_extension AND NOT EXISTS (\n\t\tSELECT * FROM pg_catalog.pg_extension\n\t\tWHERE extname = 'postgis_raster')\n\tTHEN\n\t\t\tfullver = fullver || ' [UNPACKAGED!]';\n\tEND IF;\nEND IF;\n\nIF rast_scr_ver IS NOT NULL AND rast_scr_ver != relproc THEN\n\tfullver = fullver || ' (raster procs from \"' || rast_scr_ver || '\" need upgrade)';\nEND IF;\n\nIF sfcgal_scr_ver IS NOT NULL AND sfcgal_scr_ver != relproc THEN\n\tfullver = fullver || ' (sfcgal procs from

\'" || sfcgal_scr_ver || '\" need upgrade)';\n\tEND IF;\n\n\t-- Check for the presence of deprecated functions\n\tIF EXISTS (SELECT oid FROM pg_catalog.pg_proc WHERE proname LIKE '%_deprecated_by_postgis_%')\n\tTHEN\n\t\tfullver = fullver || ' (deprecated functions exist, upgrade is not complete)';\n\tEND IF;\n\n\tRETURN fullver;\nEND\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_geos_noop",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_geos_noop(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$GEOSnoop$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_geos_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_geos_version()\n RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3', $function$postgis_geos_version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_getbbox",
    "arguments": "geometry",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_getbbox(geometry)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_to_BOX2DF$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_hasbbox",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_hasbbox(geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_hasBBOX$function$\n"
  },
  {
    "schema": "public",

```
    "function_name": "postgis_index_supportfn",
    "arguments": "internal",
    "return_type": "internal",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.postgis_index_supportfn(internal)\n RETURNS internal\n LANGUAGE c\nAS
'$libdir/postgis-3', $function$postgis_index_supportfn$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_lib_build_date",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_lib_build_date()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_lib_build_date$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_lib_revision",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_lib_revision()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_lib_revision$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_lib_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_lib_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_lib_version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_libjson_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_libjson_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$postgis_libjson_version$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "postgis_liblwgeom_version",
   "arguments": "",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.postgis_liblwgeom_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_liblwgeom_version$function$\n"
 },
 {
   "schema": "public",
   "function_name": "postgis_libprotobuf_version",
   "arguments": "",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.postgis_libprotobuf_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE STRICT\nAS '$libdir/postgis-3',
$function$postgis_libprotobuf_version$function$\n"
 },
 {
   "schema": "public",
   "function_name": "postgis_libxml_version",
   "arguments": "",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.postgis_libxml_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_libxml_version$function$\n"
 },
 {
   "schema": "public",
   "function_name": "postgis_noop",
   "arguments": "geometry",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.postgis_noop(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_noop$function$\n"
 },
 {
   "schema": "public",
   "function_name": "postgis_proj_version",
   "arguments": "",
   "return_type": "text",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_proj_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_proj_version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_scripts_build_date",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_scripts_build_date()\n
RETURNS text\n LANGUAGE sql\n IMMUTABLE\nAS $function$SELECT '2024-09-05
22:13:41'::text AS version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_scripts_installed",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_scripts_installed()\n
RETURNS text\n LANGUAGE sql\n IMMUTABLE\nAS $function$ SELECT trim('3.3.7'::text ||
$rev$ a0c7967 $rev$) AS version $function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_scripts_released",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_scripts_released()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_scripts_released$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_svn_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_svn_version()\n
RETURNS text\n LANGUAGE sql\n IMMUTABLE\nAS $function$\n\tSELECT
public._postgis_deprecate(\n\t\t'postgis_svn_version', 'postgis_lib_revision',
'3.1.0');\n\tSELECT public.postgis_lib_revision();\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_transform_geometry",
```

```
    "arguments": "geom geometry, text, text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.postgis_transform_geometry(geom geometry, text, text, integer)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$transform_geom$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_type_name",
    "arguments": "geomname character varying, coord_dimension integer, use_new_name
boolean DEFAULT true",
    "return_type": "character varying",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_type_name(geomname
character varying, coord_dimension integer, use_new_name boolean DEFAULT true)\n
RETURNS character varying\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS $function$\n\tSELECT CASE WHEN $3 THEN new_name ELSE
old_name END As geomname\n\tFROM\n\t( VALUES\n\t\t\t('GEOMETRY', 'Geometry',
2),\n\t\t\t('GEOMETRY', 'GeometryZ', 3),\n\t\t\t('GEOMETRYM', 'GeometryM',
3),\n\t\t\t('GEOMETRY', 'GeometryZM', 4),\n\n\t\t\t('GEOMETRYCOLLECTION',
'GeometryCollection', 2),\n\t\t\t('GEOMETRYCOLLECTION', 'GeometryCollectionZ',
3),\n\t\t\t('GEOMETRYCOLLECTIONM', 'GeometryCollectionM',
3),\n\t\t\t('GEOMETRYCOLLECTION', 'GeometryCollectionZM', 4),\n\n\t\t\t('POINT', 'Point',
2),\n\t\t\t('POINT', 'PointZ', 3),\n\t\t\t('POINTM','PointM', 3),\n\t\t\t('POINT', 'PointZM',
4),\n\n\t\t\t('MULTIPOINT','MultiPoint', 2),\n\t\t\t('MULTIPOINT','MultiPointZ',
3),\n\t\t\t('MULTIPOINTM','MultiPointM', 3),\n\t\t\t('MULTIPOINT','MultiPointZM',
4),\n\n\t\t\t('POLYGON', 'Polygon', 2),\n\t\t\t('POLYGON', 'PolygonZ', 3),\n\t\t\t('POLYGONM',
'PolygonM', 3),\n\t\t\t('POLYGON', 'PolygonZM', 4),\n\n\t\t\t('MULTIPOLYGON',
'MultiPolygon', 2),\n\t\t\t('MULTIPOLYGON', 'MultiPolygonZ', 3),\n\t\t\t('MULTIPOLYGONM',
'MultiPolygonM', 3),\n\t\t\t('MULTIPOLYGON', 'MultiPolygonZM',
4),\n\n\t\t\t('MULTILINESTRING', 'MultiLineString', 2),\n\t\t\t('MULTILINESTRING',
'MultiLineStringZ', 3),\n\t\t\t('MULTILINESTRINGM', 'MultiLineStringM',
3),\n\t\t\t('MULTILINESTRING', 'MultiLineStringZM', 4),\n\n\t\t\t('LINESTRING', 'LineString',
2),\n\t\t\t('LINESTRING', 'LineStringZ', 3),\n\t\t\t('LINESTRINGM', 'LineStringM',
3),\n\t\t\t('LINESTRING', 'LineStringZM', 4),\n\n\t\t\t('CIRCULARSTRING', 'CircularString',
2),\n\t\t\t('CIRCULARSTRING', 'CircularStringZ', 3),\n\t\t\t('CIRCULARSTRINGM',
'CircularStringM' ,3),\n\t\t\t('CIRCULARSTRING', 'CircularStringZM',
4),\n\n\t\t\t('COMPOUNDCURVE', 'CompoundCurve', 2),\n\t\t\t('COMPOUNDCURVE',
'CompoundCurveZ', 3),\n\t\t\t('COMPOUNDCURVEM', 'CompoundCurveM',
3),\n\t\t\t('COMPOUNDCURVE', 'CompoundCurveZM', 4),\n\n\t\t\t('CURVEPOLYGON',
'CurvePolygon', 2),\n\t\t\t('CURVEPOLYGON', 'CurvePolygonZ',
3),\n\t\t\t('CURVEPOLYGONM', 'CurvePolygonM', 3),\n\t\t\t('CURVEPOLYGON',
'CurvePolygonZM', 4),\n\n\t\t\t('MULTICURVE', 'MultiCurve', 2),\n\t\t\t('MULTICURVE',
'MultiCurveZ', 3),\n\t\t\t('MULTICURVEM', 'MultiCurveM', 3),\n\t\t\t('MULTICURVE',
'MultiCurveZM', 4),\n\n\t\t\t('MULTISURFACE', 'MultiSurface', 2),\n\t\t\t('MULTISURFACE',
'MultiSurfaceZ', 3),\n\t\t\t('MULTISURFACEM', 'MultiSurfaceM', 3),\n\t\t\t('MULTISURFACE',
```

'MultiSurfaceZM', 4),\n\n\t\t\t('POLYHEDRALSURFACE', 'PolyhedralSurface',
2),\n\t\t\t('POLYHEDRALSURFACE', 'PolyhedralSurfaceZ',
3),\n\t\t\t('POLYHEDRALSURFACEM', 'PolyhedralSurfaceM',
3),\n\t\t\t('POLYHEDRALSURFACE', 'PolyhedralSurfaceZM', 4),\n\n\t\t\t('TRIANGLE',
'Triangle', 2),\n\t\t\t('TRIANGLE', 'TriangleZ', 3),\n\t\t\t('TRIANGLEM', 'TriangleM',
3),\n\t\t\t('TRIANGLE', 'TriangleZM', 4),\n\n\t\t\t('TIN', 'Tin', 2),\n\t\t\t('TIN', 'TinZ',
3),\n\t\t\t('TINM', 'TinM', 3),\n\t\t\t('TIN', 'TinZM', 4))\n\t\t\t As g(old_name, new_name,
coord_dimension)\n\tWHERE (upper(old_name) = upper($1) OR upper(new_name) =
upper($1))\n\t\tAND coord_dimension = $2;\n$function$\n"
 },
 {
  "schema": "public",
  "function_name": "postgis_typmod_dims",
  "arguments": "integer",
  "return_type": "integer",
  "function_type": "function",
  "definition": "CREATE OR REPLACE FUNCTION public.postgis_typmod_dims(integer)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$postgis_typmod_dims$function$\n"
 },
 {
  "schema": "public",
  "function_name": "postgis_typmod_srid",
  "arguments": "integer",
  "return_type": "integer",
  "function_type": "function",
  "definition": "CREATE OR REPLACE FUNCTION public.postgis_typmod_srid(integer)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$postgis_typmod_srid$function$\n"
 },
 {
  "schema": "public",
  "function_name": "postgis_typmod_type",
  "arguments": "integer",
  "return_type": "text",
  "function_type": "function",
  "definition": "CREATE OR REPLACE FUNCTION public.postgis_typmod_type(integer)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$postgis_typmod_type$function$\n"
 },
 {
  "schema": "public",
  "function_name": "postgis_version",
  "arguments": "",
  "return_type": "text",
  "function_type": "function",

```
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_version()\n RETURNS
text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "postgis_wagyu_version",
    "arguments": "",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.postgis_wagyu_version()\n
RETURNS text\n LANGUAGE c\n IMMUTABLE\nAS '$libdir/postgis-3',
$function$postgis_wagyu_version$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec",
    "arguments": "sparsevec, integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec(sparsevec, integer,
boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_cmp",
    "arguments": "sparsevec, sparsevec",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_cmp(sparsevec,
sparsevec)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_cmp$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_eq",
    "arguments": "sparsevec, sparsevec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_eq(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_eq$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_ge",
    "arguments": "sparsevec, sparsevec",
```

```
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_ge(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_ge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_gt",
    "arguments": "sparsevec, sparsevec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_gt(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_gt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_in",
    "arguments": "cstring, oid, integer",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_in(cstring, oid,
integer)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_l2_squared_distance",
    "arguments": "sparsevec, sparsevec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.sparsevec_l2_squared_distance(sparsevec, sparsevec)\n RETURNS double
precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$sparsevec_l2_squared_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_le",
    "arguments": "sparsevec, sparsevec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_le(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_le$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "sparsevec_lt",
    "arguments": "sparsevec, sparsevec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_lt(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_lt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_ne",
    "arguments": "sparsevec, sparsevec",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_ne(sparsevec,
sparsevec)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_ne$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_negative_inner_product",
    "arguments": "sparsevec, sparsevec",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.sparsevec_negative_inner_product(sparsevec, sparsevec)\n RETURNS double
precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$sparsevec_negative_inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_out",
    "arguments": "sparsevec",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_out(sparsevec)\n
RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$sparsevec_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_recv",
    "arguments": "internal, oid, integer",
    "return_type": "sparsevec",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_recv(internal, oid,
integer)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_recv$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_send",
    "arguments": "sparsevec",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_send(sparsevec)\n
RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$sparsevec_send$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_to_halfvec",
    "arguments": "sparsevec, integer, boolean",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_to_halfvec(sparsevec,
integer, boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_to_vector",
    "arguments": "sparsevec, integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_to_vector(sparsevec,
integer, boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$sparsevec_to_vector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "sparsevec_typmod_in",
    "arguments": "cstring[]",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.sparsevec_typmod_in(cstring[])\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$sparsevec_typmod_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "spheroid_in",
    "arguments": "cstring",
```

```
    "return_type": "spheroid",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.spheroid_in(cstring)\n RETURNS spheroid\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$ellipsoid_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "spheroid_out",
    "arguments": "spheroid",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.spheroid_out(spheroid)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$ellipsoid_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dclosestpoint",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dclosestpoint(geom1 geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_closestpoint3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3ddfullywithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3ddfullywithin(geom1 geometry, geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3', $function$LWGEOM_dfullywithin3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3ddistance",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3ddistance(geom1 geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_3DDistance$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_3ddwithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3ddwithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS
'$libdir/postgis-3', $function$LWGEOM_dwithin3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dintersects",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dintersects(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$ST_3DIntersects$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dlength",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dlength(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_length_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dlineinterpolatepoint",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_3dlineinterpolatepoint(geometry, double precision)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_3DLineInterpolatePoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dlongestline",
    "arguments": "geom1 geometry, geom2 geometry",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dlongestline(geom1
geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_longestline3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dmakebox",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dmakebox(geom1 geometry,
geom2 geometry)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$BOX3D_construct$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dmaxdistance",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dmaxdistance(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_maxdistance3d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dperimeter",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dperimeter(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_perimeter_poly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_3dshortestline",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_3dshortestline(geom1
geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_shortestline3d$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_addmeasure",
    "arguments": "geometry, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_addmeasure(geometry, double
precision, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$ST_AddMeasure$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_addpoint",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_addpoint(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_addpoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_addpoint",
    "arguments": "geom1 geometry, geom2 geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_addpoint(geom1 geometry,
geom2 geometry, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_addpoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_affine",
    "arguments": "geometry, double precision, double precision, double precision, double
precision, double precision, double precision, double precision, double precision, double
precision, double precision, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_affine(geometry, double
precision, double precision, double precision, double precision, double precision, double
precision, double precision, double precision, double precision, double precision, double
precision, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_affine$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "st_affine",
   "arguments": "geometry, double precision, double precision, double precision, double
precision, double precision, double precision",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_affine(geometry, double
precision, double precision, double precision, double precision, double precision, double
precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS $function$SELECT public.ST_Affine($1,  $2, $3, 0,  $4, $5, 0,  0, 0,
1,  $6, $7, 0)$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_angle",
   "arguments": "line1 geometry, line2 geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_angle(line1 geometry, line2
geometry)\n RETURNS double precision\n LANGUAGE sql\n IMMUTABLE PARALLEL
SAFE STRICT COST 50\nAS $function$SELECT ST_Angle(St_StartPoint($1),
ST_EndPoint($1), St_StartPoint($2), ST_EndPoint($2))$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_angle",
   "arguments": "pt1 geometry, pt2 geometry, pt3 geometry, pt4 geometry DEFAULT
'0101000000000000000000F87F000000000000F87F'::geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_angle(pt1 geometry, pt2
geometry, pt3 geometry, pt4 geometry DEFAULT
'0101000000000000000000F87F000000000000F87F'::geometry)\n RETURNS double
precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_angle$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_area",
   "arguments": "geog geography, use_spheroid boolean DEFAULT true",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_area(geog geography,
use_spheroid boolean DEFAULT true)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$geography_area$function$\n"
 },
```

```
 {
   "schema": "public",
   "function_name": "st_area",
   "arguments": "text",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_area(text)\n RETURNS double
precision\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$
SELECT public.ST_Area($1::public.geometry);  $function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_area",
   "arguments": "geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_area(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$ST_Area$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_area2d",
   "arguments": "geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_area2d(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$ST_Area$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_asbinary",
   "arguments": "geometry, text",
   "return_type": "bytea",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_asbinary(geometry, text)\n
RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_asBinary$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_asbinary",
   "arguments": "geography, text",
   "return_type": "bytea",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_asbinary(geography, text)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_asBinary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asbinary",
    "arguments": "geography",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asbinary(geography)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_asBinary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asbinary",
    "arguments": "geometry",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asbinary(geometry)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_asBinary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asencodedpolyline",
    "arguments": "geom geometry, nprecision integer DEFAULT 5",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asencodedpolyline(geom geometry, nprecision integer DEFAULT 5)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_asEncodedPolyline$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkb",
    "arguments": "geometry, text",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkb(geometry, text)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$WKBFromLWGEOM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkb",
```

```
    "arguments": "geometry",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkb(geometry)\n
RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$WKBFromLWGEOM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkt",
    "arguments": "geography, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkt(geography, integer)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asEWKT$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkt",
    "arguments": "geometry, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkt(geometry, integer)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asEWKT$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkt",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkt(text)\n RETURNS
text\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$
SELECT public.ST_AsEWKT($1::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asewkt",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkt(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asEWKT$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "st_asewkt",
    "arguments": "geography",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asewkt(geography)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asEWKT$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgeojson",
    "arguments": "geom geometry, maxdecimaldigits integer DEFAULT 9, options integer
DEFAULT 8",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgeojson(geom geometry,
maxdecimaldigits integer DEFAULT 9, options integer DEFAULT 8)\n RETURNS text\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_asGeoJson$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgeojson",
    "arguments": "r record, geom_column text DEFAULT ''::text, maxdecimaldigits integer
DEFAULT 9, pretty_bool boolean DEFAULT false",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgeojson(r record,
geom_column text DEFAULT ''::text, maxdecimaldigits integer DEFAULT 9, pretty_bool
boolean DEFAULT false)\n RETURNS text\n LANGUAGE c\n STABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_AsGeoJsonRow$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgeojson",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgeojson(text)\n RETURNS
text\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$
SELECT public.ST_AsGeoJson($1::public.geometry, 9, 0);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgeojson",
    "arguments": "geog geography, maxdecimaldigits integer DEFAULT 9, options integer
DEFAULT 0",
```

```
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgeojson(geog geography,
maxdecimaldigits integer DEFAULT 9, options integer DEFAULT 0)\n RETURNS text\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_as_geojson$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgml",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgml(text)\n RETURNS text\n
LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$
SELECT public._ST_AsGML(2,$1::public.geometry,15,0, NULL, NULL);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgml",
    "arguments": "geog geography, maxdecimaldigits integer DEFAULT 15, options integer
DEFAULT 0, nprefix text DEFAULT 'gml'::text, id text DEFAULT ''::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgml(geog geography,
maxdecimaldigits integer DEFAULT 15, options integer DEFAULT 0, nprefix text DEFAULT
'gml'::text, id text DEFAULT ''::text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_as_gml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgml",
    "arguments": "version integer, geog geography, maxdecimaldigits integer DEFAULT 15,
options integer DEFAULT 0, nprefix text DEFAULT 'gml'::text, id text DEFAULT ''::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgml(version integer, geog
geography, maxdecimaldigits integer DEFAULT 15, options integer DEFAULT 0, nprefix text
DEFAULT 'gml'::text, id text DEFAULT ''::text)\n RETURNS text\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_as_gml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgml",
    "arguments": "geom geometry, maxdecimaldigits integer DEFAULT 15, options integer
DEFAULT 0",
```

```
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgml(geom geometry,
maxdecimaldigits integer DEFAULT 15, options integer DEFAULT 0)\n RETURNS text\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_asGML$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asgml",
    "arguments": "version integer, geom geometry, maxdecimaldigits integer DEFAULT 15,
options integer DEFAULT 0, nprefix text DEFAULT NULL::text, id text DEFAULT NULL::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asgml(version integer, geom
geometry, maxdecimaldigits integer DEFAULT 15, options integer DEFAULT 0, nprefix text
DEFAULT NULL::text, id text DEFAULT NULL::text)\n RETURNS text\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_asGML$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ashexewkb",
    "arguments": "geometry, text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ashexewkb(geometry, text)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_asHEXEWKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ashexewkb",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ashexewkb(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_asHEXEWKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_askml",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_askml(text)\n RETURNS text\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$ SELECT public.ST_AsKML($1::public.geometry, 15);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_askml",
    "arguments": "geom geometry, maxdecimaldigits integer DEFAULT 15, nprefix text DEFAULT ''::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_askml(geom geometry, maxdecimaldigits integer DEFAULT 15, nprefix text DEFAULT ''::text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_asKML$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_askml",
    "arguments": "geog geography, maxdecimaldigits integer DEFAULT 15, nprefix text DEFAULT ''::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_askml(geog geography, maxdecimaldigits integer DEFAULT 15, nprefix text DEFAULT ''::text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$geography_as_kml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_aslatlontext",
    "arguments": "geom geometry, tmpl text DEFAULT ''::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_aslatlontext(geom geometry, tmpl text DEFAULT ''::text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_to_latlon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_asmarc21",
    "arguments": "geom geometry, format text DEFAULT 'hdddmmss'::text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asmarc21(geom geometry, format text DEFAULT 'hdddmmss'::text)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_AsMARC21$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_asmvtgeom",
    "arguments": "geom geometry, bounds box2d, extent integer DEFAULT 4096, buffer
integer DEFAULT 256, clip_geom boolean DEFAULT true",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_asmvtgeom(geom geometry,
bounds box2d, extent integer DEFAULT 4096, buffer integer DEFAULT 256, clip_geom
boolean DEFAULT true)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE COST 500\nAS '$libdir/postgis-3', $function$ST_AsMVTGeom$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_assvg",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_assvg(text)\n RETURNS text\n
LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$
SELECT public.ST_AsSVG($1::public.geometry,0,15);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_assvg",
    "arguments": "geog geography, rel integer DEFAULT 0, maxdecimaldigits integer
DEFAULT 15",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_assvg(geog geography, rel
integer DEFAULT 0, maxdecimaldigits integer DEFAULT 15)\n RETURNS text\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_as_svg$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_assvg",
    "arguments": "geom geometry, rel integer DEFAULT 0, maxdecimaldigits integer
DEFAULT 15",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_assvg(geom geometry, rel
integer DEFAULT 0, maxdecimaldigits integer DEFAULT 15)\n RETURNS text\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_asSVG$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "st_astext",
    "arguments": "geography, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_astext(geography, integer)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asText$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_astext",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_astext(geometry)\n RETURNS
text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asText$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_astext",
    "arguments": "geometry, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_astext(geometry, integer)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asText$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_astext",
    "arguments": "geography",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_astext(geography)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_asText$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_astext",
    "arguments": "text",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_astext(text)\n RETURNS text\n
LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$
SELECT public.ST_AsText($1::public.geometry);  $function$\n"
```

```
 },
 {
   "schema": "public",
   "function_name": "st_astwkb",
   "arguments": "geom geometry[], ids bigint[], prec integer DEFAULT NULL::integer, prec_z
integer DEFAULT NULL::integer, prec_m integer DEFAULT NULL::integer, with_sizes
boolean DEFAULT NULL::boolean, with_boxes boolean DEFAULT NULL::boolean",
   "return_type": "bytea",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_astwkb(geom geometry[], ids
bigint[], prec integer DEFAULT NULL::integer, prec_z integer DEFAULT NULL::integer,
prec_m integer DEFAULT NULL::integer, with_sizes boolean DEFAULT NULL::boolean,
with_boxes boolean DEFAULT NULL::boolean)\n RETURNS bytea\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3',
$function$TWKBFromLWGEOMArray$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_astwkb",
   "arguments": "geom geometry, prec integer DEFAULT NULL::integer, prec_z integer
DEFAULT NULL::integer, prec_m integer DEFAULT NULL::integer, with_sizes boolean
DEFAULT NULL::boolean, with_boxes boolean DEFAULT NULL::boolean",
   "return_type": "bytea",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_astwkb(geom geometry, prec
integer DEFAULT NULL::integer, prec_z integer DEFAULT NULL::integer, prec_m integer
DEFAULT NULL::integer, with_sizes boolean DEFAULT NULL::boolean, with_boxes boolean
DEFAULT NULL::boolean)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE COST 50\nAS '$libdir/postgis-3', $function$TWKBFromLWGEOM$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_asx3d",
   "arguments": "geom geometry, maxdecimaldigits integer DEFAULT 15, options integer
DEFAULT 0",
   "return_type": "text",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_asx3d(geom geometry,
maxdecimaldigits integer DEFAULT 15, options integer DEFAULT 0)\n RETURNS text\n
LANGUAGE sql\n IMMUTABLE PARALLEL SAFE COST 500\nAS $function$SELECT
public._ST_AsX3D(3,$1,$2,$3,'');$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_azimuth",
   "arguments": "geog1 geography, geog2 geography",
   "return_type": "double precision",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_azimuth(geog1 geography,
geog2 geography)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_azimuth$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_azimuth",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_azimuth(geom1 geometry,
geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_azimuth$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_bdmpolyfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_bdmpolyfromtext(text,
integer)\n RETURNS geometry\n LANGUAGE plpgsql\n IMMUTABLE PARALLEL SAFE
STRICT\nAS $function$\nDECLARE\n\tgeomtext alias for $1;\n\tsrid alias for $2;\n\tmline
public.geometry;\n\tgeom public.geometry;\nBEGIN\n\tmline :=
public.ST_MultiLineStringFromText(geomtext, srid);\n\n\tIF mline IS
NULL\n\tTHEN\n\t\tRAISE EXCEPTION 'Input is not a MultiLinestring';\n\tEND IF;\n\n\tgeom
:= public.ST_Multi(public.ST_BuildArea(mline));\n\n\tRETURN geom;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_bdpolyfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_bdpolyfromtext(text, integer)\n
RETURNS geometry\n LANGUAGE plpgsql\n IMMUTABLE PARALLEL SAFE STRICT\nAS
$function$\nDECLARE\n\tgeomtext alias for $1;\n\tsrid alias for $2;\n\tmline
public.geometry;\n\tgeom public.geometry;\nBEGIN\n\tmline :=
public.ST_MultiLineStringFromText(geomtext, srid);\n\n\tIF mline IS
NULL\n\tTHEN\n\t\tRAISE EXCEPTION 'Input is not a MultiLinestring';\n\tEND IF;\n\n\tgeom
:= public.ST_BuildArea(mline);\n\n\tIF public.GeometryType(geom) !=
'POLYGON'\n\tTHEN\n\t\tRAISE EXCEPTION 'Input returns more then a single polygon, try
using BdMPolyFromText instead';\n\tEND IF;\n\n\tRETURN geom;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_boundary",
```

```
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_boundary(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$boundary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_boundingdiagonal",
    "arguments": "geom geometry, fits boolean DEFAULT false",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_boundingdiagonal(geom geometry, fits boolean DEFAULT false)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$ST_BoundingDiagonal$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_box2dfromgeohash",
    "arguments": "text, integer DEFAULT NULL::integer",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_box2dfromgeohash(text, integer DEFAULT NULL::integer)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS '$libdir/postgis-3', $function$box2d_from_geohash$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "geom geometry, radius double precision, quadsegs integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(geom geometry, radius double precision, quadsegs integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS $function$ SELECT public.ST_Buffer($1, $2, CAST('quad_segs='||CAST($3 AS text) as text)) $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "text, double precision",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(text, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$ SELECT public.ST_Buffer($1::public.geometry, $2);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "geography, double precision, text",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(geography, double precision, text)\n RETURNS geography\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$SELECT public.geography(public.ST_Transform(public.ST_Buffer(public.ST_Transform(public.geometry($1), public._ST_BestSRID($1)), $2, $3), public.ST_SRID($1)))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "text, double precision, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(text, double precision, integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$ SELECT public.ST_Buffer($1::public.geometry, $2, $3); $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "text, double precision, text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(text, double precision, text)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$ SELECT public.ST_Buffer($1::public.geometry, $2, $3); $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "geography, double precision, integer",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(geography, double precision, integer)\n RETURNS geography\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$SELECT
```

```
public.geography(public.ST_Transform(public.ST_Buffer(public.ST_Transform(public.geomet
ry($1), public._ST_BestSRID($1)), $2, $3), public.ST_SRID($1)))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "geography, double precision",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(geography, double
precision)\n RETURNS geography\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE
STRICT\nAS $function$SELECT
public.geography(public.ST_Transform(public.ST_Buffer(public.ST_Transform(public.geomet
ry($1), public._ST_BestSRID($1)), $2), public.ST_SRID($1)))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buffer",
    "arguments": "geom geometry, radius double precision, options text DEFAULT ''::text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buffer(geom geometry, radius
double precision, options text DEFAULT ''::text)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$buffer$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_buildarea",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_buildarea(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_BuildArea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_centroid",
    "arguments": "geography, use_spheroid boolean DEFAULT true",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_centroid(geography,
use_spheroid boolean DEFAULT true)\n RETURNS geography\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_centroid$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "st_centroid",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_centroid(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$centroid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_centroid",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_centroid(text)\n RETURNS
geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS $function$
SELECT public.ST_Centroid($1::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_chaikinsmoothing",
    "arguments": "geometry, integer DEFAULT 1, boolean DEFAULT false",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_chaikinsmoothing(geometry,
integer DEFAULT 1, boolean DEFAULT false)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_ChaikinSmoothing$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_cleangeometry",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_cleangeometry(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_CleanGeometry$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_clipbybox2d",
    "arguments": "geom geometry, box box2d",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_clipbybox2d(geom geometry,
box box2d)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_ClipByBox2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_closestpoint",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_closestpoint(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_closestpoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_closestpointofapproach",
    "arguments": "geometry, geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_closestpointofapproach(geometry, geometry)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_ClosestPointOfApproach$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_clusterintersecting",
    "arguments": "geometry[]",
    "return_type": "geometry[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_clusterintersecting(geometry[])\n RETURNS geometry[]\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$clusterintersecting_garray$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_clusterwithin",
    "arguments": "geometry[], double precision",
    "return_type": "geometry[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_clusterwithin(geometry[],
double precision)\n RETURNS geometry[]\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3',
$function$cluster_within_distance_garray$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "st_collect",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_collect(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_collect$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_collect",
    "arguments": "geometry[]",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_collect(geometry[])\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_collect_garray$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_collectionextract",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_collectionextract(geometry,
integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$ST_CollectionExtract$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_collectionextract",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_collectionextract(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_CollectionExtract$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_collectionhomogenize",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_collectionhomogenize(geometry)\n RETURNS geometry\n LANGUAGE c\n
```

IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_CollectionHomogenize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_combinebbox",
    "arguments": "box2d, geometry",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_combinebbox(box2d,
geometry)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE\nAS
'$libdir/postgis-3', $function$BOX2D_combine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_combinebbox",
    "arguments": "box3d, geometry",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_combinebbox(box3d,
geometry)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST
50\nAS '$libdir/postgis-3', $function$BOX3D_combine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_combinebbox",
    "arguments": "box3d, box3d",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_combinebbox(box3d, box3d)\n
RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS
'$libdir/postgis-3', $function$BOX3D_combine_BOX3D$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_concavehull",
    "arguments": "param_geom geometry, param_pctconvex double precision,
param_allow_holes boolean DEFAULT false",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_concavehull(param_geom
geometry, param_pctconvex double precision, param_allow_holes boolean DEFAULT
false)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS '$libdir/postgis-3', $function$ST_ConcaveHull$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_contains",

```
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_contains(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$contains$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_containsproperly",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_containsproperly(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS
'$libdir/postgis-3', $function$containsproperly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_convexhull",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_convexhull(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$convexhull$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_coorddim",
    "arguments": "geometry geometry",
    "return_type": "smallint",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_coorddim(geometry
geometry)\n RETURNS smallint\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_ndims$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_coveredby",
    "arguments": "geog1 geography, geog2 geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_coveredby(geog1 geography,
geog2 geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
```

```
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$geography_coveredby$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_coveredby",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_coveredby(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$coveredby$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_coveredby",
    "arguments": "text, text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_coveredby(text, text)\n
RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$
SELECT public.ST_CoveredBy($1::public.geometry, $2::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_covers",
    "arguments": "geog1 geography, geog2 geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_covers(geog1 geography,
geog2 geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$geography_covers$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_covers",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_covers(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$covers$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_covers",
    "arguments": "text, text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_covers(text, text)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$ SELECT public.ST_Covers($1::public.geometry, $2::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_cpawithin",
    "arguments": "geometry, geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_cpawithin(geometry, geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_CPAWithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_crosses",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_crosses(geom1 geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3', $function$crosses$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_curvetoline",
    "arguments": "geom geometry, tol double precision DEFAULT 32, toltype integer DEFAULT 0, flags integer DEFAULT 0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_curvetoline(geom geometry, tol double precision DEFAULT 32, toltype integer DEFAULT 0, flags integer DEFAULT 0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_CurveToLine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_delaunaytriangles",
    "arguments": "g1 geometry, tolerance double precision DEFAULT 0.0, flags integer DEFAULT 0",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_delaunaytriangles(g1
geometry, tolerance double precision DEFAULT 0.0, flags integer DEFAULT 0)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_DelaunayTriangles$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dfullywithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dfullywithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS
'$libdir/postgis-3', $function$LWGEOM_dfullywithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_difference",
    "arguments": "geom1 geometry, geom2 geometry, gridsize double precision DEFAULT
'-1.0'::numeric",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_difference(geom1 geometry,
geom2 geometry, gridsize double precision DEFAULT '-1.0'::numeric)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_Difference$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dimension",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dimension(geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_dimension$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_disjoint",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_disjoint(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$disjoint$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "st_distance",
   "arguments": "geom1 geometry, geom2 geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_distance(geom1 geometry,
geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_Distance$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_distance",
   "arguments": "text, text",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_distance(text, text)\n
RETURNS double precision\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE
STRICT\nAS $function$ SELECT public.ST_Distance($1::public.geometry,
$2::public.geometry);  $function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_distance",
   "arguments": "geog1 geography, geog2 geography, use_spheroid boolean DEFAULT
true",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_distance(geog1 geography,
geog2 geography, use_spheroid boolean DEFAULT true)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$geography_distance$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_distancecpa",
   "arguments": "geometry, geometry",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_distancecpa(geometry,
geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_DistanceCPA$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_distancesphere",
   "arguments": "geom1 geometry, geom2 geometry, radius double precision",
   "return_type": "double precision",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_distancesphere(geom1
geometry, geom2 geometry, radius double precision)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_distance_sphere$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_distancesphere",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_distancesphere(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE sql\n
IMMUTABLE PARALLEL SAFE STRICT\nAS $function$select public.ST_distance(
public.geography($1), public.geography($2),false)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_distancespheroid",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_distancespheroid(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_distance_ellipsoid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_distancespheroid",
    "arguments": "geom1 geometry, geom2 geometry, spheroid",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_distancespheroid(geom1
geometry, geom2 geometry, spheroid)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_distance_ellipsoid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dump",
    "arguments": "geometry",
    "return_type": "SETOF geometry_dump",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dump(geometry)\n RETURNS
SETOF geometry_dump\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$LWGEOM_dump$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_dumppoints",
    "arguments": "geometry",
    "return_type": "SETOF geometry_dump",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dumppoints(geometry)\n
RETURNS SETOF geometry_dump\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$LWGEOM_dumppoints$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dumprings",
    "arguments": "geometry",
    "return_type": "SETOF geometry_dump",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dumprings(geometry)\n
RETURNS SETOF geometry_dump\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_dump_rings$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dumpsegments",
    "arguments": "geometry",
    "return_type": "SETOF geometry_dump",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dumpsegments(geometry)\n
RETURNS SETOF geometry_dump\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_dumpsegments$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dwithin",
    "arguments": "text, text, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dwithin(text, text, double
precision)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS
$function$ SELECT public.ST_DWithin($1::public.geometry, $2::public.geometry, $3);
$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dwithin",
    "arguments": "geog1 geography, geog2 geography, tolerance double precision,
use_spheroid boolean DEFAULT true",
```

```
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dwithin(geog1 geography,
geog2 geography, tolerance double precision, use_spheroid boolean DEFAULT true)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$geography_dwithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_dwithin",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_dwithin(geom1 geometry,
geom2 geometry, double precision)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS
'$libdir/postgis-3', $function$LWGEOM_dwithin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_endpoint",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_endpoint(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_endpoint_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_envelope",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_envelope(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_envelope$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_equals",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_equals(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
```

STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$ST_Equals$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_estimatedextent",
    "arguments": "text, text, text, boolean",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_estimatedextent(text, text, text,
boolean)\n RETURNS box2d\n LANGUAGE c\n STABLE STRICT SECURITY
DEFINER\nAS '$libdir/postgis-3', $function$gserialized_estimated_extent$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_estimatedextent",
    "arguments": "text, text, text",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_estimatedextent(text, text,
text)\n RETURNS box2d\n LANGUAGE c\n STABLE STRICT SECURITY DEFINER\nAS
'$libdir/postgis-3', $function$gserialized_estimated_extent$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_estimatedextent",
    "arguments": "text, text",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_estimatedextent(text, text)\n
RETURNS box2d\n LANGUAGE c\n STABLE STRICT SECURITY DEFINER\nAS
'$libdir/postgis-3', $function$gserialized_estimated_extent$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(geometry, double
precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "geom geometry, dx double precision, dy double precision, dz double
precision DEFAULT 0, dm double precision DEFAULT 0",

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(geom geometry, dx
double precision, dy double precision, dz double precision DEFAULT 0, dm double precision
DEFAULT 0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "box box3d, dx double precision, dy double precision, dz double precision
DEFAULT 0",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(box box3d, dx double
precision, dy double precision, dz double precision DEFAULT 0)\n RETURNS box3d\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$BOX3D_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "box3d, double precision",
    "return_type": "box3d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(box3d, double
precision)\n RETURNS box3d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$BOX3D_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "box box2d, dx double precision, dy double precision",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(box box2d, dx double
precision, dy double precision)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT\nAS '$libdir/postgis-3', $function$BOX2D_expand$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_expand",
    "arguments": "box2d, double precision",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_expand(box2d, double
precision)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$BOX2D_expand$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_exteriorring",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_exteriorring(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_exteriorring_polygon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_filterbym",
    "arguments": "geometry, double precision, double precision DEFAULT NULL::double
precision, boolean DEFAULT false",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_filterbym(geometry, double
precision, double precision DEFAULT NULL::double precision, boolean DEFAULT false)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_FilterByM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_findextent",
    "arguments": "text, text, text",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_findextent(text, text, text)\n
RETURNS box2d\n LANGUAGE plpgsql\n STABLE PARALLEL SAFE STRICT\nAS
$function$\nDECLARE\n\tschemaname alias for $1;\n\ttablename alias for
$2;\n\tcolumnname alias for $3;\n\tmyrec RECORD;\nBEGIN\n\tFOR myrec IN EXECUTE
'SELECT public.ST_Extent(\"' || columnname || '\") As extent FROM \"' || schemaname || '\".\"'
|| tablename || '\"' LOOP\n\t\treturn myrec.extent;\n\tEND LOOP;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_findextent",
    "arguments": "text, text",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_findextent(text, text)\n
RETURNS box2d\n LANGUAGE plpgsql\n STABLE PARALLEL SAFE STRICT\nAS
$function$\nDECLARE\n\ttablename alias for $1;\n\tcolumnname alias for $2;\n\tmyrec
RECORD;\n\nBEGIN\n\tFOR myrec IN EXECUTE 'SELECT public.ST_Extent(\"' ||
columnname || '\") As extent FROM \"' || tablename || '\"' LOOP\n\t\treturn
myrec.extent;\n\tEND LOOP;\nEND;\n$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_flipcoordinates",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_flipcoordinates(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_FlipCoordinates$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_force2d",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_force2d(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_force_2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_force3d",
    "arguments": "geom geometry, zvalue double precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_force3d(geom geometry, zvalue double precision DEFAULT 0.0)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Force3DZ($1, $2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_force3dm",
    "arguments": "geom geometry, mvalue double precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_force3dm(geom geometry, mvalue double precision DEFAULT 0.0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_force_3dm$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_force3dz",
    "arguments": "geom geometry, zvalue double precision DEFAULT 0.0",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_force3dz(geom geometry,
zvalue double precision DEFAULT 0.0)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_force_3dz$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_force4d",
    "arguments": "geom geometry, zvalue double precision DEFAULT 0.0, mvalue double
precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_force4d(geom geometry,
zvalue double precision DEFAULT 0.0, mvalue double precision DEFAULT 0.0)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_force_4d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcecollection",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcecollection(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_force_collection$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcecurve",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcecurve(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$LWGEOM_force_curve$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcepolygonccw",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcepolygonccw(geometry)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$ SELECT public.ST_Reverse(public.ST_ForcePolygonCW($1))
$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_forcepolygoncw",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcepolygoncw(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_force_clockwise_poly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcerhr",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcerhr(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_force_clockwise_poly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcesfs",
    "arguments": "geometry, version text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcesfs(geometry, version
text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_force_sfs$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_forcesfs",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_forcesfs(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$LWGEOM_force_sfs$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_frechetdistance",
    "arguments": "geom1 geometry, geom2 geometry, double precision DEFAULT
'-1'::integer",
    "return_type": "double precision",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_frechetdistance(geom1 geometry, geom2 geometry, double precision DEFAULT '-1'::integer)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_FrechetDistance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_fromflatgeobuf",
    "arguments": "anyelement, bytea",
    "return_type": "SETOF anyelement",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_fromflatgeobuf(anyelement, bytea)\n RETURNS SETOF anyelement\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3', $function$pgis_fromflatgeobuf$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_fromflatgeobuftotable",
    "arguments": "text, text, bytea",
    "return_type": "void",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_fromflatgeobuftotable(text, text, bytea)\n RETURNS void\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$pgis_tablefromflatgeobuf$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_generatepoints",
    "arguments": "area geometry, npoints integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_generatepoints(area geometry, npoints integer)\n RETURNS geometry\n LANGUAGE c\n PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_GeneratePoints$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_generatepoints",
    "arguments": "area geometry, npoints integer, seed integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_generatepoints(area geometry, npoints integer, seed integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_GeneratePoints$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_geogfromtext",
    "arguments": "text",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geogfromtext(text)\n
RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$geography_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geogfromwkb",
    "arguments": "bytea",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geogfromwkb(bytea)\n
RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$geography_from_binary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geographyfromtext",
    "arguments": "text",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geographyfromtext(text)\n
RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$geography_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geohash",
    "arguments": "geog geography, maxchars integer DEFAULT 0",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geohash(geog geography,
maxchars integer DEFAULT 0)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_GeoHash$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geohash",
    "arguments": "geom geometry, maxchars integer DEFAULT 0",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geohash(geom geometry,
maxchars integer DEFAULT 0)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_GeoHash$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "st_geomcollfromtext",
   "arguments": "text, integer",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_geomcollfromtext(text,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 500\nAS $function$\n\tSELECT CASE\n\tWHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) =
'GEOMETRYCOLLECTION'\n\tTHEN public.ST_GeomFromText($1,$2)\n\tELSE NULL
END\n\t$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_geomcollfromtext",
   "arguments": "text",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_geomcollfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE\n\tWHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'GEOMETRYCOLLECTION'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_geomcollfromwkb",
   "arguments": "bytea",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_geomcollfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE\n\tWHEN
public.geometrytype(public.ST_GeomFromWKB($1)) =
'GEOMETRYCOLLECTION'\n\tTHEN public.ST_GeomFromWKB($1)\n\tELSE NULL
END\n\t$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_geomcollfromwkb",
   "arguments": "bytea, integer",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_geomcollfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE\n\tWHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) =
```

'GEOMETRYCOLLECTION'\n\tTHEN public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geometricmedian",
    "arguments": "g geometry, tolerance double precision DEFAULT NULL::double precision, max_iter integer DEFAULT 10000, fail_if_not_converged boolean DEFAULT false",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geometricmedian(g geometry, tolerance double precision DEFAULT NULL::double precision, max_iter integer DEFAULT 10000, fail_if_not_converged boolean DEFAULT false)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 10000\nAS '$libdir/postgis-3', $function$ST_GeometricMedian$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geometryfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geometryfromtext(text, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geometryfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geometryfromtext(text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geometryn",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geometryn(geometry, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_geometryn_collection$function$\n"
  },
  {
    "schema": "public",

```
    "function_name": "st_geometrytype",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geometrytype(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$geometry_geometrytype$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromewkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromewkb(bytea)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOMFromEWKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromewkt",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromewkt(text)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$parse_WKT_lwgeom$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgeohash",
    "arguments": "text, integer DEFAULT NULL::integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgeohash(text,
integer DEFAULT NULL::integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE
PARALLEL SAFE COST 50\nAS $function$ SELECT
CAST(public.ST_Box2dFromGeoHash($1, $2) AS geometry); $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgeojson",
    "arguments": "jsonb",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgeojson(jsonb)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$SELECT public.ST_GeomFromGeoJson($1::text)$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgeojson",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgeojson(text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$geom_from_geojson$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgeojson",
    "arguments": "json",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgeojson(json)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$SELECT public.ST_GeomFromGeoJson($1::text)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgml",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgml(text, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$geom_from_gml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromgml",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromgml(text)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$SELECT public._ST_GeomFromGML($1, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromkml",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromkml(text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$geom_from_kml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfrommarc21",
    "arguments": "marc21xml text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfrommarc21(marc21xml text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_GeomFromMARC21$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromtext(text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromtext(text, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromtwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromtwkb(bytea)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOMFromTWKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromwkb",
    "arguments": "bytea, integer",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$SELECT public.ST_SetSRID(public.ST_GeomFromWKB($1),
$2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_geomfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_geomfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_from_WKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_gmltosql",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_gmltosql(text, integer)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$geom_from_gml$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_gmltosql",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_gmltosql(text)\n RETURNS
geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
$function$SELECT public._ST_GeomFromGML($1, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_hasarc",
    "arguments": "geometry geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_hasarc(geometry geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_has_arc$function$\n"
  },
  {
```

```
    "schema": "public",
    "function_name": "st_hausdorffdistance",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_hausdorffdistance(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$hausdorffdistance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_hausdorffdistance",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_hausdorffdistance(geom1
geometry, geom2 geometry, double precision)\n RETURNS double precision\n LANGUAGE
c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$hausdorffdistancedensify$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_hexagon",
    "arguments": "size double precision, cell_i integer, cell_j integer, origin geometry
DEFAULT '0101000000000000000000000000000000000000'::geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_hexagon(size double
precision, cell_i integer, cell_j integer, origin geometry DEFAULT
'0101000000000000000000000000000000000000'::geometry)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_Hexagon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_hexagongrid",
    "arguments": "size double precision, bounds geometry, OUT geom geometry, OUT i
integer, OUT j integer",
    "return_type": "SETOF record",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_hexagongrid(size double
precision, bounds geometry, OUT geom geometry, OUT i integer, OUT j integer)\n
RETURNS SETOF record\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$ST_ShapeGrid$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_interiorringn",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_interiorringn(geometry,
integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_interiorringn_polygon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_interpolatepoint",
    "arguments": "line geometry, point geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_interpolatepoint(line geometry,
point geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$ST_InterpolatePoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersection",
    "arguments": "text, text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersection(text, text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS $function$ SELECT public.ST_Intersection($1::public.geometry,
$2::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersection",
    "arguments": "geom1 geometry, geom2 geometry, gridsize double precision DEFAULT
'-1'::integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersection(geom1 geometry,
geom2 geometry, gridsize double precision DEFAULT '-1'::integer)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_Intersection$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersection",
    "arguments": "geography, geography",
    "return_type": "geography",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersection(geography,
geography)\n RETURNS geography\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE
STRICT\nAS $function$SELECT
public.geography(public.ST_Transform(public.ST_Intersection(public.ST_Transform(public.g
eometry($1), public._ST_BestSRID($1, $2)), public.ST_Transform(public.geometry($2),
public._ST_BestSRID($1, $2))), public.ST_SRID($1)))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersects",
    "arguments": "text, text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersects(text, text)\n
RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$
SELECT public.ST_Intersects($1::public.geometry, $2::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersects",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersects(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$ST_Intersects$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_intersects",
    "arguments": "geog1 geography, geog2 geography",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_intersects(geog1 geography,
geog2 geography)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$geography_intersects$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isclosed",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isclosed(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_isclosed$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_iscollection",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_iscollection(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$ST_IsCollection$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isempty",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isempty(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_isempty$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ispolygonccw",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ispolygonccw(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_IsPolygonCCW$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ispolygoncw",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ispolygoncw(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_IsPolygonCW$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isring",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_isring(geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$isring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_issimple",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_issimple(geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$issimple$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvalid",
    "arguments": "geometry, integer",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvalid(geometry, integer)\n RETURNS boolean\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS $function$SELECT (public.ST_isValidDetail($1, $2)).valid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvalid",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvalid(geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$isvalid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvaliddetail",
    "arguments": "geom geometry, flags integer DEFAULT 0",
    "return_type": "valid_detail",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvaliddetail(geom geometry, flags integer DEFAULT 0)\n RETURNS valid_detail\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$isvaliddetail$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvalidreason",
```

```
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvalidreason(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$isvalidreason$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvalidreason",
    "arguments": "geometry, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvalidreason(geometry,
integer)\n RETURNS text\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS $function$\n\tSELECT CASE WHEN valid THEN 'Valid Geometry' ELSE
reason END FROM (\n\t\tSELECT (public.ST_isValidDetail($1, $2)).*\n\t) foo\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_isvalidtrajectory",
    "arguments": "geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_isvalidtrajectory(geometry)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_IsValidTrajectory$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_length",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_length(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_length2d_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_length",
    "arguments": "geog geography, use_spheroid boolean DEFAULT true",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_length(geog geography,
use_spheroid boolean DEFAULT true)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_length$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_length",
    "arguments": "text",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_length(text)\n RETURNS
double precision\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS
$function$ SELECT public.ST_Length($1::public.geometry);  $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_length2d",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_length2d(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_length2d_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_length2dspheroid",
    "arguments": "geometry, spheroid",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_length2dspheroid(geometry,
spheroid)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_length2d_ellipsoid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_lengthspheroid",
    "arguments": "geometry, spheroid",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_lengthspheroid(geometry,
spheroid)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_length_ellipsoid_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_letters",
    "arguments": "letters text, font json DEFAULT NULL::json",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_letters(letters text, font json
DEFAULT NULL::json)\n RETURNS geometry\n LANGUAGE plpgsql\n IMMUTABLE
PARALLEL SAFE COST 500\n SET standard_conforming_strings TO 'on'\nAS
$function$\nDECLARE\n  letterarray text[];\n  letter text;\n  geom geometry;\n  prevgeom
geometry = NULL;\n  adjustment float8 = 0.0;\n  position float8 = 0.0;\n  text_height float8 =
100.0;\n  width float8;\n  m_width float8;\n  spacing float8;\n  dist float8;\n  wordarr
geometry[];\n  wordgeom geometry;\n  -- geometry has been run through
replace(encode(st_astwkb(geom),'base64'), E'\\n', '')\n  font_default_height float8 = 1000.0;\n
font_default json = '{\n
\"!\":\"BgACAQhUrgsTFOQCABQAExELiwi5AgAJiggBYQmJCgAOAg4CDAIOBAoEDAYKBg
oGCggICAgICAgGCgYKBgoGCgQMBAoECgQMAgoADAIKAAoADAEKAAwBCgMKAQwDC
gMKAwoFCAUKBwgHBgcIBwYJBgkECwYJBAsCDQILAg0CDQANAQ0BCwELAwsDCwUJ
BQkFCQcHBwcHBwcFCQUJBQkFCQMLAwkDCQMLAQkACwEJAAkACwIJAAsCCQQJAgs
ECQQJBAkGBwYJCAcIBQgHCAUKBQoDDAUKAQwDDgEMAQ4BDg==\",\n
\"&\":\"BgABAskBygP+BowEAACZAmcAANsCAw0FDwUNBQ0FDQcLBw0HCwcLCQsJCwk
LCQkJCwsJCwkLCQ0HCwcNBw8HDQUPBQ8DDwMRAw8DEQERARERBEQERABcAFQIX
AhUCEwQVBBMGEwYTBhEIEQgPChEKDwoPDA0MDQwNDgsOCRAJEAkQBxAHEgUSB
RQFFAMUAxQBFgEWARgAigEAFAISABICEgQQAhAEEAQQBg4GEAoOCg4MDg4ODgwS
DgsMCwoJDAcMBwwFDgUMAw4DDgEOARABDgEQARIBEAASAHgAIAQeBB4GHAgaCh
oMGA4WDhYQFBISEhISDhQQFAwWDBYKFgoYBhgIGAQYBBgCGgAaABgBGAMYAxYHF
gUWCRYJFAsUCxIPEg0SERARDhMOFQwVDBcIGQYbBhsCHQIfAR+dAgAADAAKAQoBC
gEIAwgFBgUGBQYHBAUEBwQHAgcCBwIHAAcABwAHAQcBBwMHAwUDBwUFBQUHBQ
UBBwMJAQkBCQAJAJcBAAUCBQAFAgUEBQIDBAUEAwQDBgMEAQYDBgEGAAgBBgA
KSeECAJ8BFi84HUQDQCAAmAKNAQAvExMx\",\n
\"\\\"\":\"BgACAQUmwguEAgAAkwSDAgAAIAQBBfACAIACAACTBP8BAACUBA==\",\n
\"'\":\"BgABAQUmwguEAgAAkwSDAgAAIAQ=\",\n
\"(\":\"BgABAUOQBNwLDScNKw0rCysLLwsxCTEJMwc1BzcHNwM7AzsDPwE/AEEANwI1Aj
MEMwIzBjEGLwYvCC0ILQgrCCkKKQonCicMJbkCAAkqCSoHLAksBywFLgcuBS4FMAMwA
zADMgEwATQBMgA0ADwCOgI6BDoEOAY4BjYINgg2CjQKMgoyCjIMMAwwDi7AAgA=\",\n
\")\":\"BgABAUMQ3Au6AgAOLQwvDC8KMQoxCjEKMwg1CDUGNQY3BDcEOQI5AjkAOwA
zATEBMQExAy8DLwMvBS8FLQctBS0HKwktBykJKwkpswIADCYKKAooCioIKggsCC4ILgY
wBjAGMgQ0AjQCNAI2ADgAQgFAAz4DPAM8BzgHOAc2CTQJMgsyCzALLg0sDSoNKg==\"
,\n
\"+\":\"BgABAQ3IBOwGALcBuAEAANUBtwEAALcB0wEAALgBtwEAANYBuAEAALgB1AEA\
",\n  \"/\":\"BgABAQVCAoIDwAuyAgCFA78LrQIA\",\n
\"4\":\"BgABAhDkBr4EkgEAEREApwJ/AADxARIR5QIAEhIA9AHdAwAA7ALIA9AG6gIAERE
A8QYFqwIAAIIDwwH/AgABxAEA\",\n
\"v\":\"BgABASDmA5AEPu4CROwBExb6AgAZFdMC0wgUFaECABIU0wLWCBcW+AIAExV
E6wEEFQQXBBUEFwQVBBUEFwQVBBUEFwQVBBUEFwQXBBUEFwYA\",\n
\",\":\"BgABAWMYpAEADgIOAgwCDgQMBAoGDAYKBgoICAgICAgICAoGCgYKBAoEDAQK
BAoCDAIKAgwCCgAKAAwACgEMAQoBCgMMAwoDCgUKBQgFCgUIBwYJCAcGCQYJBA
sGCQQLAg0CCwINAg0AAwABAAMAAwADAQMAAwADAAMBBQAFAQcBBwEHAwcBCQ
MJAQsDCwMLAw0FDQMNBQ8FDwURBxMFEwkTBxcJFwkXswEAIMgBCQYJBgkGBwYJC
AcIBQgHCgUKBQoFDAEMAwwBDgEOABA=\",\n
\"-\":\"BgABAQUq0AMArALEBAAAqwLDBAA=\",\n
\".\":\"BgABAWFOrAEADgIOAg4CDgQMBAoGDAYKBgoICAgKCAgIBgoGCgYKBgoEDAQK
BAwECgIMAAwCDAAMAAwBCgAMAQoDDAMKAwoDCgUKBQgFCgUIBwgJBgcICQYJBgs
```

GCQQLAg0CDQINAA0ADQENAQ0BCwMNAwkFCwUJBQkHBwcJBwUHBwkFCQUJBQkD
CwMJAwsDCQELAAsBCwALAAsCCQALAgkECwQJBAkECQYJBgcGBwgJBgcKBQgHCgU
KBQwFCgEOAwwBDgEOAA4=\",\n
\"0\":\"BgABAoMB+APaCxwAHAEaARoDFgMYBRYFFAcUBxIJEgkQCRALEAsOCwwNDA0
MDQoPCg0IDwgPBhEGDwYRBA8EEQIRAhMCEQITABMA4QUAEQETAREBEQMRAxEFE
QURBREHDwkPBw8JDwsNCw0LDQ0NDQsNCw8JEQkRCREJEwcTBxUFFQUVAxUDFwE
XARkAGQAZAhcCFwQXBBUGEwYTCBMIEQoRCg8KDwoPDA0MDQ4NDgsOCQ4JEAkQ
BxAHEAUSBRIDEgMSAxIDEgESARQAEgDiBQASAhQCEgISBBIEEgYSBhIGEggQChAIEA
oQDBAMDgwODg4ODA4MEgwQChIKEggUCBQIFgYWBBYGGAQYAhgCGgILZIcDHTZBE
kMRHTUA4QUeOUITRBIePADiBQ==\",\n
\"2\":\"BgABAWpUwALUA44GAAoBCAEKAQgDBgMGBQYFBgUEBwQFBAUCBwIHAgUAB
wAHAAUBBwMFAQcFBQMHBQUHBQcFBwMJAwkBCQELAQsAC68CAAAUAhIAFAISBBQ
CEgQUBBIEEgYUCBIGEAgSChAKEAoQDBAMDg4ODgwQDBIMEgoSChQIFggWCBgGGA
QaAhwCHAIWABQBFgEUARQDFAMSAxQFEgUSBxIHEAkQCRALDgsODQ4NDA8KDwwR
CBMKEwgTBhUGFwQXBBcEGwAbABsAHQEftwPJBdIDAACpAhIPzwYAFBIArgI=\",\n
\"1\":\"BgABARCsBLALAJ0LEhERADcA2QEANwATABQSAOYIpwEAALgCERKEBAASABE
R\",\n
\"3\":\"BgABAZ0B/gbEC/sB0QQOAwwBDAMMAwwFCgMKBQoFCgUIBwoFCAcICQgJBgkI
CQYLCAsECwYLBA0GDwINBA8CDwQRAhECEQITABUCFQAVAH0AEQETAREBEQETAx
EDEQURBREFDwcRBw8JDwkNCQ8LDQsNDQsNCw0LDwsPCREJEQcRBxMFFQUVBRU
DFwEXARkAGQAZAhkCFwQVBBUEEwYTCBEIEQgRCg0MDwoNDA0OCw4LDgkQCRAH
EAkQBRAFEgUSAxIDFAMSAxYBFAEWARYAFqQCAAALAgkCCQQHAgcGBwYHBgUIBQY
DCAMIAwYDCAEIAQgACAAIAAgCCAIIAgYCCAQIBAgGBgYEBgQIBAoCCgAKAAwAvAEA
BgEIAAYBBgMGAwQDBgMEBQQDBAUCBQQFAgUABwIFAJkBAACmAaIB3ALbAgAREQD
mAhIRggYA\",\n
\"5\":\"BgABAaAB0APgBxIAFAESABIBEgMSARADEgMQAxIFEAcOBRAHDgkOCQ4JDgsM
CwwLCgsKDQoPCA0IDwgPBhEEEwYTAhMEFwIXABcAiQIAEwETABEBEQMTAxEDDwMR
BQ8FDwUPBw8JDQcNCQ0LDQsLCwsNCw0JDwkPCREHEQcTBxMFEwMVAxcDGQEZAR
kAFwAVAhUCFQQTBBMGEwYRCBEIDwoPCg8KDQwNDA0MCw4LDgkOCRAJEAcOBxAH
EgUQBRIDEAMSAxIBEgEUARIAFLgCAAAFAgUABQIFBAUCBQQDBAUEAwYDBgMIAwg
BCAEIAQoACAAIAgYACAQGAgQEBgQEBAQGBAQCBgIGAgYCBgIIAAYA4AEABgEIAAYB
BgMGAQQDBgMEAwQFBAMCBQQFAgUABwIFAPkBAG+OAQCCBRESAgAAAuYFABMR
AK8CjQMAAJ8BNgA=\",\n
\"7\":\"BgABAQrQBsILhQOvCxQR7wIAEhK+AvYIiwMAAKgCERKwBgA=\",\n
\"6\":\"BgABAsYBnAOqBxgGFgYYBBYEFgIWABQBFgEUAxQDFAUUBRIFEAcSCRAJEAkO
Cw4NDgsMDQoPCg8KDwgRCBEGEQYRBBMCEwITAhUAkwIBAAERARAREBEQEPAxEFEQ
MPBREFDwcPBw8HDwkNCQ0LDQsNCwsNCw0LDQkPCQ8JDwcRBxEHEwUTAxMFFQE
XAxcBGQAVABUCEwIVBBMEEQYTBhEIEQgPChEKDQoPDA0MDQwNDgsOCxALDgkQC
RAHEgcQBxIFEgUSBRIBFAMSARIBFAASAOIFABACEgIQAhIEEAQQBhIGEAYQCBAKEA
gOChAMDgwMDA4ODA4MDgwODBAKEAoQChIIEggSBhQGFgYUAhYCGAIYABoAGAEY
ARYBFgMUBRQFEgUSBxAHEAcQCQ4LDgkMCwwNDA0KDQgPCg0GEQgPBhEEEQQRB
BMEEwITAhMCFQIVABWrAgAACgEIAQoBCAEGAwYDBgUGBQQFBAUEBQQFAgUABwIF
AAUABwEFAAUBBQMFAwUDBQMFBQMFAwUBBQEHAQkBBwAJAJcBDUbpBDASFi4A4
AETLC8SBQAvERUrAN8BFC0yEQQA\",\n
\"8\":\"BgABA9gB6gPYCxYAFAEUARYBEgMUBRQFEgUSBxIHEAcSCQ4JEAkOCw4LDgs
MDQwNCg0KDQoPCg8IDwgPBhEGEQQPBBMCEQIRABMAQwAxAA8BEQEPAREDDwM
RAw8FEQUPBxEJDwkPCQ8NDw0PDQ8IBwYHCAcGBwgHBgkGBwYJBgcECQYJBAkGC
QQJBAsECwQLBA0CCwINAg8CDwIPAA8AaQATAREBEwERAxEFEQURBREHEQcPBw8J

DwkPCw8LDQsNDQ0LCw0LDwsNCQ8JDwcPBw8HEQURAxEFEQMRARMBEwFDABEAE
wIRAhEEEQQRBg8GEQgPCA8KDwoPCg0MDQwNDAsOCw4LDgkQCRAJDgkQBxIHEAcS
BRADEgMUAxIBFAEUABQAagAOAhAADgIOAg4EDAIOBAwEDAQMBgwECgYMBAoGCA
YKBgoGCggKBgoICgYICAoICA0MCwwLDgsOCRAHEAcQBxIFEgUSAxIDEgMSARABEgA
SADIARAASAhICEgQSAhIGEAYSBhAIEAgQCBAKDgoODA4MDgwMDgwODA4KEAwQCB
IKEggSCBQIFAYUBBQEFgQWAhYCGAANT78EFis0EwYANBIYLgC0ARcsMRQFADERGS
0AswELogHtAhcuNxA3DRkvALMBGjE6ETYSGDIAtAE=\",\n
\"9\":\"BgABAsYBpASeBBcFFQUXAxUDFQEVABMCFQITBBMEEwYRBhMGDwgRCg8KDw
oNDA0OCwwNDgkQCRAJEAcSBxIFEgUSAxQBFAEUARYAIAICAAISAhICEgQSAhAGEgQ
QBhIGEAgSCA4IEAoOChAMDAwODAwODA4MEAoOChAKEAgSCBIIFAYUBBQGFgIYBBg
CGgAWABYBFAEWAxQDEgUUBRIHEgcQCRIJEAkOCw4LDgsODQwNDA0MDwoPCg8ID
wgRCBEGEQYRBhEEEQITAhECEwARAOEFAA8BEQEPAREDDwMPBREFDwUPBw8JDw
cNCQ8LDQsLCw0NCw0LDQsNCw8JEQkPCREHEQcTBRMFEwUTARUBFQEXABkAFwIX
AhcCFQQTBhMGEQYRCA8IDwgNCg8MCwoLDAsOCQ4JDgkQBxAHEAUQBRIFEgMSAx
QDFAEUAxQAFgEWABamAgAACwIJAgkCCQIHBAcEBwYFBgUGAwYDBgMGAQgBBgEIA
AgABgIIAgYCBgQGBAYEBgYGBgQIBAgECAIKAgoCCgAMAJgBDUXqBC8RFS0A3wEUKz
ARBgAwEhYsAOABEy4xEgMA\",\n
\":\":\"BgACAWE0rAEADgIOAg4CDgQMBAoGDAYKBgoICAgKCAgIBgoGCgYKBgoEDAQK
BAwECgIMAAwCDAAMAAwBCgAMAQoDDAMKAwoDCgUKBQgFCgUIBwgJBgcICQYJBgs
GCQQLAg0CDQINAA0ADQENAQ0BCwMNAwkFCwUJBQkHBwcJBwUHBwkFCQUJBQkD
CwMJAwsDCQELAAsBCwALAAsCCQALAgkECwQJBAkECQYJBgcGBwgJBgcKBQgHCgU
KBQwFCgEOAwwBDgEOAA4BYQDqBAAOAg4CDgIOBAwECgYMBgoGCggICAoICAgGCg
YKBgoGCgQMBAoEDAQKAgwADAIMAAwADAEKAAwBCgMMAwoDCgMKBQoFCAUKBQ
gHCAkGBwgJBgkGCwYJBAsCDQINAg0ADQANAQ0BDQELAw0DCQULBQkFCQcHBwkH
BQcHCQUJBQkFCQMLAwkDCwEJAwsACwELAAsACwIJAAsECQILBAkECQQJBgkGBwY
HCAkGBwoFCAcKBQoFDAUKAQ4DDAEOAQ4ADg==\",\n
\"x\":\"BgABARHmAoAJMIMBNLUBNrYBMIQB1AIA9QG/BI4CvwTVAgA5hgFBwAFFxwE1fd
UCAI4CwATzAcAE1AIA\",\n
\";\":\"BgACAWEslgYADgIOAg4CDgQMBAoGDAYKBgoICAgKCAgIBgoGCgYKBgoEDAQKB
AwECgIMAAwCDAAMAAwBCgAMAQoDDAMKAwoDCgUKBQgFCgUIBwgJBgcICQYJBgsG
CQQLAg0CDQINAA0ADQENAQ0BCwMNAwkFCwUJBQkHBwcJBwUHBwkFCQUJBQkDC
wMJAwsBCQMLAAsBCwALAAsCCQALBAkCCwQJBAkECQYJBgcGBwgJBgcKBQgHCgU
KBQwFCgEOAwwBDgEOAA4BYwjxBAAOAg4CDAIOBAwECgYMBgoGCggICAgICAgICgY
KBgoECgQMBAoECgIMAgoCDAIKAAoADAAKAQwBCgEKAwwDCgMKBQoFCAUKBQgHB
gkIBwYJBgkECwYJBAsCDQILAg0CDQADAAEAAwADAAMBAwADAAMAAwEFAAUBBwE
HAQcDBwEJAwkBCwMLAwsDDQUNAw0FDwUPBREHEwUTCRMHFwkXCRezAQAgyAEJ
BgkGCQYHBgkIBwgFCAcKBQoFCgUMAQwDDAEOAQ4AEA==\",\n
\"=\":\"BgACAQUawAUA5gHEBAAA5QHDBAABBQC5AgDsAcQEAADrAcMEAA==\",\n
\"B\":\"BgABA2e2BMQLFgAUARQBFAEUAxIDEgUSBRIFEAcQBxAJDgkOCQ4LDgsMCww
NDA0KDQgNCg0IDwYPBg8GDwQRBBEEEQIRAhMAEwAHAAkABwEHAAkBCQAHAQkB
CQEHAQkBCQMJAwcDCQMJAwkFBwUJAwkHCQUHBQkHCQcJBwcHBwkHBwcJBwsHC
QUQBQ4FDgcOCQ4JDAkMCwoNCg0IDwgRBhMEFQQXAhcCGwDJAQEvAysFJwkIDSMP
HREbFRkXFRsTHw8fCyUJJwcrAy0B6wMAEhIAoAsREuYDAAiRAYEEIgEAKioSSA1EOR6J
AQAA0wEJkAGPBSwSEiwAzAETKikSjwEAAMUCkAEA\",\n
\"A\":\"BgABAg/KBfIBqQIAN98BEhHzAgAWEuwCngsREvwCABMR8gKdCxIR8QIAFBI54AE
FIwGCBk3TA6ABAE3UAwMA\",\n
\"?\":\"BgACAe4BsgaYCAAZABkBFwEXBRUDEwUTBxEHEQcPCQ8JDQkNCQ0LCwsLCws
LCQsJCwcNBwsHDQcLBQsFDQULAwkFCwMLAwkDCQMBAAABAQABAAEBAQABAAEA

AQABAAABAQAAAQEAEwcBAQABAAMBAwADAAUABQAFAAcABwAFAAcABwAFAgcAB
QAHAAUAW7cCAABcABgBFgAUAhQAFAISAhACEAIQBA4EDgQMBgwGDAYMBgoICgYK
CAgKCggICAgKBgoICgYMCAwGDAgOBg4GEAYQBgIAAgIEAAICBAACAgQCBAIKBAoGC
AQKBggIBgYICAYIBggGCgQIBAoECAQKAggCCgIKAAgACgAKAAgBCAEKAwgDCAMIAw
gFBgMIBQYHBAUGBQQFBAcCBQQHAgcCCQIHAgkCBwAJAgkACQAJAAkBCQAJAQsAC
QELAQsDCwELAwsDCwMLAwsDCwULAwsFCwMLBV2YAgYECAQKBAwGDAQMBhAIEA
YSBhIIEgYUBhIEFgYUBBYEFgQWAhgCFgIYABYAGAAYARgBGAMWBRYHFgcWCRYLFA
0IBQYDCAUIBwYFCAcGBwgHBgcICQYJCAkGCQYJCAsGCwYLBgsGDQYNBA0GDQQN
BA8EDwQPAg8EEQIRAhEAEQITAWGpBesGAA4CDgIOAg4EDAQKBgwGCgYKCAgICggI
CAYKBgoGCgYKBAwECgQMBAoCDAAMAgwADAAMAQoADAEKAwwDCgMKAwoFCgUIB
QoFCAcICQYHCAkGCQYLBgkECwINAg0CDQANAA0BDQENAQsDDQMJBQsFCQUJBwc
HCQcFBwcJBQkFCQUJAwsDCQMLAwkBCwALAQsACwALAgkACwIJBAsECQQJBAkGCQ
YHBgcICQYHCgUIBwoFCgUMBQoBDgMMAQ4BDgAO\",\n
\"C\":\"BgABAWmmA4ADAAUCBQAFAgUEBQIDBAUEAwQDBgMEAQYDBgEGAAgBBgD
WAgAAwQLVAgATABMCEQITBBEEEQQRBhEIEQgPCA8KDwoNCg0MDQwNDAsOCw4LD
gkOCxAHEAkQBxIHEgUSBRIDEgEUARIBFAAUAMIFABQCFAISBBQEEgQSBhIIEggSCBA
KEAoQCg4MDgwODA4ODA4MDgwQDA4KEggQChIIEggSBhIGFAQSAhQCEgIUAMYCAA
DBAsUCAAUABwEFAAUBBQMDAQUDAwMDAwMFAQMDBQEFAAUBBwAFAMEF\",\n
\"L\":\"BgABAQmcBhISEdkFABIQALQLwgIAAIEJ9AIAAK8C\",\n
\"D\":\"BgABAkeyBMQLFAAUARIBFAESAxIDEgMSBRIFEAcQBxAHDgkOCQ4LDgsMCwwN
DA0KDwoPCg8IDwgRCBEGEwQTBBMEEwIVAhUAFwDBBQAXARcBFwMTAxUDEwUTBx
EHEQcPCQ8JDwkNCw0LCwsLDQsNCQ0JDQcPBw8HDwcRBREFEQMRAxEDEwERARM
BEwDfAwASEgCgCxES4AMACT6BAxEuKxKLAQAAvwaMAQAsEhIsAMIF\",\n
\"F\":\"BgABARGABoIJ2QIAAIECsgIAEhIA4QIRErECAACvBBIR5QIAEhIAsgucBQASEgDlA
hES\",\n
\"E\":\"BgABARRkxAuWBQAQEgDlAhES0QIAAP0BtgIAEhIA5wIRFLUCAAD/AfACABISAO
UCERLDBQASEgCyCw==\",\n
\"G\":\"BgABAZsBjgeIAgMNBQ8FDQUNBQ0HCwcNBwsHCwkLCQsJCwsJCwsLCQsJDQkL
Bw0HDwcNBw8FDwUPAw8DEQMPAxEBEQERARMBEQAXABUCFwIVAhMEFQQTBhMG
EwYRCBEIDwoRCg8KDwwNDA0MDQ4LDgkQCRAJEAcQBxIFEgUUBRQDFAMUARYBFg
EYAMoFABQCFAASBBQCEgQSBBIEEgYSBhAGEAgQCBAKDgoOCg4MDgwMDgwOChA
KEAoSCBIIFAgUBhQEGAYWAhgEGAIaAOoCAAC3AukCAAcABwEFAQUBBQMFAwMFAw
UDBQEFAQcBBQEFAQUABwAFAMUFAAUCBwIFAgUCBQQFBAMGBQYDBgUGAwgDBg
MIAQgDCAEIAQoBCAEIAAgACgAIAAgCCAIIAggECgQGBAgECAYIBgC6AnEAAJwCmAM
AAJcF\",\n
\"H\":\"BgABARbSB7ILAQAAnwsSEeUCABISAOAE5QEAAN8EEhHlAgASEgCiCxEQ5gIAE
REA/QPmAQAAgAQPEOYCABER\",\n
\"I\":\"BgABAQmuA7ILAJ8LFBHtAgAUEgCgCxMS7gIAExE=\",\n
\"J\":\"BgABAWuqB7ILALEIABEBEwERAREDEwMRAxEFEQURBw8HEQcPCQ0LDwsNCw
0NDQ0LDwsPCxEJEQkTCRMJFQcVBxcFFwMZAxsBGwEbAB8AHQIbAhsEGQYXBhcGFQ
gTCBMKEwoRDA8KDwwNDA0OCw4LDgkQCRAJEAcQBRIFEgUSAxQDEgESARIBFAESA
BIAgAEREtoCABERAn8ACQIHBAcEBwYHBgUIBQoDCgMKAwoDDAEKAQwBCgEMAAwA
CgAMAgoCDAIKBAoECgYKBggGBgYGCAQGBAgCCgAIALIIERLmAgAREQ==\",\n
\"M\":\"BgACAQRm1gsUABMAAAABE5wIAQDBCxIR5QIAEhIA6gIK5gLVAe0B1wHuAQztAg
DhAhIR5QIAEhIAxAsUAPoDtwT4A7YEFgA=\",\n
\"K\":\"BgABAVXMCRoLBQsDCQMLAwsDCwMLAwsBCwELAQsBCwELAQ0ACwELAAsAD
QALAg0ACwILAA0CCwILAgsCDQQLBAsECwYNBAsGCwYLCAsGCwgJCgsICQoJCgkMC
QwJDAkOCRALEAkQCRKZAdICUQAAiwQSEecCABQSAKALExLoAgAREQC3BEIA+AG4B

AEAERKCAwAREdkCzQXGAYUDCA0KDQgJCgkMBwoFDAUMAQwBDgAMAg4CDAQOB
AwGDghmlQI=\",\n
\"O\":\"BgABAoMBsATaCxwAHAEaARoDGgMYBRYFFgcWBxQJEgkSCRILEAsODQ4NDg0
MDwoNDA8KDwgPCBEIDwYRBg8GEQQRAhMCEQITABMA0QUAEQETAREBEQMTBREF
EQURBxEHDwcRCQ8LDQsPCw0NDQ0NDwsPCw8LEQkTCRMJEwkVBxUHFwUXAxkDG
QEbARsAGwAZAhkCGQQXBhcGFQYVCBUIEwoRChEMEQoRDA8MDQ4NDg0OCxAJEAs
QCRAHEgcSBxIFFAMSAxIDEgEUARIAEgDSBQASAhQCEgISBBIEEgYSBhIIEggQCBAKE
gwODBAMEA4ODg4QDhIMEAwSChQKFAgUCBYIFgYYBBoGGgQcAh4CHgILggGLAylCW
xZbFSlBANEFKklcGVwYKkwA0gU=\",\n
\"N\":\"BgABAQ+YA/oEAOUEEhHVAgASEgC+CxQAwATnBQDIBRMS2AIAExEAzQsRAL8El
gU=\",\n
\"P\":\"BgABAkqoB5AGABcBFQEVAxMDEwMTBREHEQcRBw8JDwkNCQ0LDQsNCwsNC
w0JDQkNCQ8HDwcPBxEFEQURAxEDEQMTAREBEwETAH8AAIMDEhHlAgASEgCgCxES
1AMAFAAUARIAFAESAxIDEgMSAxIFEAUQBRAHDgkOCQ4JDgsMCwwNDA0KDQoNCg8I
DwgRCBEGEwQTBBUEFQIXAhkAGQCzAgnBAsoCESwrEn8AANUDgAEALBISLgDYAg==\
",\n
\"R\":\"BgABAj9msgsREvYDABQAFAESARQBEgESAxIDEgUSBRAFEAcQBw4JDgkOCQ4L
DAsMDQwLCg0KDwoNCA8IDwgPBhEEEwYTAhMEFQIXABcAowIAEwEVARMDEwMTBR
MFEQcTBxELEQsRDQ8PDREPEQ0VC8QB/QMSEfkCABQSiQGyA3EAALEDFBHnAgASE
gCgCwnCAscFogEALhISLACqAhEsLRKhAQAApQM=\",\n
\"Q\":\"BgABA4YBvAniAbkB8wGZAYABBQUFAwUFBQUHBQUDBwUFBQcFBQMHBQcDB
wUJAwcDCQMJAwkDCQMJAQsDCwMLAQsDCwENAw0BDQEPAA8BDwAPABsAGwIZAhc
EGQQXBBUGFQgVCBMIEQoTChEKDwwPDA8ODQ4NDgsQCxAJEAkQBxIHEgUSBRQFF
AMUARQDFAEWABYAxgUAEgIUAhICEgQSBBIGEgYSCBIIEAgQChIMDgwQDBAODg4OE
A4SDBAMEgoUChQIFAgWCBYGGAQaBhoEHAIeAh4CHAAcARoBGgMaAxgFFgUWBxYH
FAkSCRIJEgsQCw4NDg0ODQwPCg0MDwoPCA8IEQgPBhEGDwYRBBECEwIRAhMAEwC
7BdgBrwElmQSyAwC6AylAWxZbFSk/AP0BjAK7AQeLAoMCGEc4J0wHVBbvAaYBAEM=\",\
n
\"S\":\"BgABAYMC8gOEBxIFEgUQBxIFEgcSBxIJEgcSCRIJEAkQCRALEAsOCw4NDg0MD
Q4PDA0KEQoPChEKEQgRCBMGFQQTBBcCFQAXABkBEwARAREBEQMPAQ8DDwMPA
w0DDQUNAw0FCwULBwsFCwUJBwsFCQcHBQkHCQUHBwcHBwUHBwUFBQcHBwUHA
wcFEQsRCxMJEwkTBxMFEwUVBRUDFQMVARMBFwEVABUAFQIVAhUCFQQVBBUEEw
YVBhMIEwgTCBMIEwgRCBMKEQgRCmK6AgwFDgUMAw4FEAUOBRAFEAUQBRAFEAM
SAw4DEAMQAxABEAEOAQ4AEAIMAg4CDgQMBAwGCggKCAoKBgwGDgYQBBACCgAM
AAoBCAMKBQgFCAcIBwgJCAsGCQgLCA0IDQgNCA8IDQgPCA8IDwgPChEIDwgPCBEK
DwoPDBEMDwwPDg8ODw4NEA0QCxALEgsSCRIHEgcUBRQFGAUYAxgBGgEcAR4CJAY
kBiAIIAweDBwQHBAYEhgUFBYUFhQWEBoQGg4aDBwKHAoeBh4GIAQgAiACIgEiASIFIg
UiBSAJIgkgCyINZ58CBwQJAgkECwQLAgsECwINBA0CDQQNAg0CDQALAg0ADQANAAs
BCwELAQsDCwULBQkFCQcHBwcJBwkFCwMLAw0BDQENAAsCCwQLBAkGCQgJCAkKB
woJCgcMBQoHDAcMBQwF\",\n
\"V\":\"BgABARG2BM4DXrYEbKwDERL0AgAVEesCnQsSEfsCABQS8QKeCxES8gIAExFuq
wNgtQQEAA==\",\n
\"T\":\"BgABAQskxAv0BgAAtQKVAgAA+wgSEeUCABISAPwImwIAALYC\",\n
\"U\":\"BgABAW76B7ALAKMIABcBFwMXARUFFQUTBxMHEwkRCREJEQsPDQ0LDw0ND
wsPCw8LEQkPCRMJEQcTBxMFEwUVBRUDEwMXARUBFQEXABUAEwIVAhMCFQQTBB
UEEwYTBhMIEwgRChEIEQwRDA8MDw4PDg0OCxANEAsSCRIJEgcUBxQHFAMWBRYB
GAEYARgApggBAREU9AIAExMAAgCICAALAgkECQQHBAcIBwgHCAUKBQoDCgMKAww
BCgEMAQwADAAMAgoCDAIKAgoECgQKBggGCAYICAYKBAgCCgIMAgwApggAARMU9AI

AExM=\",\n
\"X\":\"BgABARmsCBISEYkDABQSS54BWYICXYkCRZUBEhGJAwAUEtYCzgXVAtIFExKIA
wATEVClAVj3AVb0AVKqAREShgMAERHXAtEF2ALNBQ==\",\n
\"W\":\"BgABARuODcQLERHpAp8LFBHlAgASEnW8A2+7AxIR6wIAFBKNA6ALERKSAwAT
EdQB7wZigARZ8AIREugCAA8RaKsDYsMDXsoDaqYDExLqAgA=\",\n
\"Y\":\"BgABARK4BcQLhgMAERHnAvMGAKsEEhHnAgAUEgCsBOkC9AYREoYDABERWO
EBUJsCUqICVtwBERI=\",\n
\"Z\":\"BgABAQmAB8QLnwOBCaADAADBAusGAMgDggmhAwAAwgLGBgA=\",\n
\"`\":\"BgABAQfqAd4JkQHmAQAOlgJCiAGpAgALiwIA\",\n
\"c\":\"BgABAW3UA84GBQAFAQUABQEFAwMBBQMDAwMDAwUBAwMFAQUABQEHAAU
AnQMABQIFAAUCBQQFAgMEBQQDBAMGAwQBBgMGAQYABgEGAPABABoMAMsCGw
7tAQATABMCEwARAhMEEQIPBBEEDwQPBg8IDwYNCA0KDQoNCgsMCwwLDAkOCRAH
DgcQBxIFEgUUBRQDFAEWAxgBGAAYAKQDABQCFAISBBQCEgYSBhAGEggQCBAIEAo
QCg4MDAwODAwODAwKDgwQCg4IEAgQCBAIEAYSBhIGEgQSAhQCFAIUAOABABwOA
M0CGQzbAQA=\",\n
\"a\":\"BgABApoB8AYCxwF+BwkHCQcJCQkHBwkHBwcJBQkFBwUJBQkFCQMHBQkDCQ
MJAwcDCQEHAQkBBwEJAQcABwAHAQcABQAHAAUBBQAFABMAEwITAhEEEwQPBBE
GDwgPCA0IDwoLCg0KCwwLDAsMCQ4JDgkOBw4HEAcQBRAFEAUSAxADEgESAxIBFA
ESABQAFAISAhQCEgQSBBIEEgYSBhIIEAgQChAIDgwODA4MDg4MDgwODBAMEAoSCB
IKEggUCBQGFgYWBBgEGAIaAhoAcgAADgEMAQoBCgEIAwgDBgUEBQQFBAcCBwIHAg
kCCQAJAKsCABcPAMwCHAvCAgAUABYBEgAUARIDFAMQAxIDEAUSBQ4FEAcOCRAJD
AkOCwwLDA0MCwoNCg8IDwgPCA8GEQYRBhMEEwIXAhUCFwAZAIMGFwAKmQLqA38
ATxchQwgnGiMwD1AMUDYAdg==\",\n
\"b\":\"BgABAkqmBIIJGAAYARYBFgEUAxQDEgUSBRIFEAcQCQ4HDgkOCw4LDAsMDQoN
Cg0KDQgPBg8GDwYRBBEEEQQTBBECEwIVAhMAFQD/AgAZARcBFwEXAxUDEwUTBR
EFEQcPBw8JDwkNCQ0LDQsLCwsNCQ0JDQcPBw8HDwURAxEDEQMTAxMBEwMVARU
AFQHPAwAUEgCWCxEY5gIAERkAowKCAQAJOvECESwrEn8AAJsEgAEALBISLgCeAw==
\",\n
\"d\":\"BgABAkryBgDLAXAREQ8NEQ0PDREJDwkRBw8FDwURAw8DDwERAw8BEQEPAC
MCHwQfCB0MGw4bEhcUFxgVGhEeDSANJAkmBSgDKgEuAIADABYCFAIUAhQCFAQUB
BIGEgYSBhAIEAgQCBAKDgoODAwMDAwMDgoOCg4KEAgQCBIGEgYSBhQEFgQWBBY
CGAIYAHwAAKQCERrmAgARFwCnCxcADOsCugJGMgDmA3sAKxERLQCfAwolHBUmBS
QKBAA=\",\n
\"e\":\"BgABAqMBigP+AgAJAgkCCQQHBAcGBwYFCAUIBQgDCgMIAQoDCAEKAQoACgA
KAAoCCAIKAggECgQIBAgGCAYGBgQIBAoECAIKAAyiAgAAGQEXARcBFwMVBRMFEwU
RBxEHDwcPCQ8LDQkNCwsNCw0LDQkNBw8JDwcPBQ8FEQURAxEDEwMTAxMBFQAVA
RcALwIrBCkIJwwlDiESHxQbGBkaFR4TIA0iCyQJKAMqASwAggMAFAIUABIEFAISBBIEEg
QSBhIGEAgQCBAIEAoODA4MDgwODgwQDBAKEAoSChIIFAgUCBYGGAQYBhoCGgQcA
h4ALgEqAygFJgkkDSANHhEaFRgXFBsSHQ4fDCUIJwQpAi0AGQEXAxcDFQcTBRMJEQk
PCw8LDQ0PDQsNDQ8LEQsRCxEJEwkTCRMJEwcTBxUHFQUVBRUHFQUVBRUHFwcV
BRUHCs4BkAMfOEUURxEfMwBvbBhAGBwaBiA=\",\n
\"h\":\"BgABAUHYBJAGAAYBBgAGAQYDBgEEAwYDBAMEBQQDAgUEBQIFAAUCBQB1A
AC5BhIT5wIAFhQAlAsRGOYCABEZAKMCeAAYABgBFgEWARQDFAMSBRIFEgUQBxAJD
gcOCQ4LDgsMCwwNCg0KDQoNCA8GDwYPBhEEEQQRBBMEEQITAhUCEwAVAO0FFhP
nAgAUEgD+BQ==\",\n
\"g\":\"BgABArkBkAeACQCNCw8ZERkRFxEVExMVERUPFQ8XDRcLGQkZBxsFGwUdAR0
BDQALAA0ADQINAAsCDQANAg0CDQILAg0EDQINBA0GDQQNBg0EDQYNCA0GDwgNC
A0IDQgPCg0KDwwNDA8MDw4PDqIB7gEQDRALEAkQCQ4JEAcOBw4FDgUOAwwFDgM

MAQwBDAEMAQwACgEKAAoACAIIAAgCCAIGAggCBgIGBAYCBgQEAgYEAqIBAQADAA
EBAwADAAMABQADAAUAAwAFAAMABQAFAAMABQA3ABMAEwIRAhMCEQQRBBEEE
QYRBg8IDwgPCA0KDQoNCg0MCwwLDgsOCQ4JDgkQBxAHEgcSBRIDFAMWAxQBFgEY
ABgA/gIAFgIWAhQEFgQUBBIGFAgSCBIIEAoSChAKDgwODA4MDg4MDgwODA4KEAgQC
BAIEgYSBhIEEgYSBBQCEgIUAhQCOgAQABABDgEQAQ4BEAMOAw4FDgUOBQwFDgc
MBQ4HDAkMB4oBUBgACbsCzQYAnAR/AC0RES0AnQMSKy4RgAEA\",\n
\"f\":\"BgABAUH8A6QJBwAHAAUABwEFAQcBBQEFAwUDBQMDAwMDAwUDAwMFAQUA
wQHCAQAWEgDZAhUUwQEAAOMEFhftAgAWFADKCQoSChIKEAoQCg4KDgwOCgwMD
AoKDAwMCgwIDAgMCAwIDAYOCAwEDgYMBA4GDAIOBA4CDgQOAg4CDgAOAg4ADgC
2AQAcDgDRAhkQowEA\",\n
\"i\":\"BgACAQIQABISALoIERLqAgAREQC5CBIR6QIAAWELyAoADgIOAgwEDgIKBgwGCg
YKCAoGCAgICggIBggGCgYKBAoECgQMBAoCDAIMAgwCDAAMAAwADAEMAQoBDAMK
AwoDCgUKBQgFCgUIBwgHCAcICQgJBgkECwQJBA0CCwANAA0ADQELAQ0BCwMJBQs
FCQUJBwkFBwcHBwcJBQcFCQUJBQkDCQMLAwkBCwELAQsACwALAAsCCwILAgkCCw
IJBAkECQQJBgcGCQYHCAcIBwgHCgUKBQwFCgMMAQwBDgEMAA4=\",\n
\"j\":\"BgACAWFKyAoADgIOAgwEDgIKBgwGCgYKCAoGCAgICggIBggGCgYKBAoECgQM
BAoCDAIMAgwCDAAMAAwADAEMAQoBDAMKAwoDCgUKBQgFCgUIBwgHCAcICQgJBg
kECwQJBA0CCwANAA0ADQELAQ0BCwMJBQsFCQUJBwkFBwcHBwcJBQcFCQUJBQkD
CQMLAwkBCwELAQsACwALAAsCCwILAgkCCwIJBAkECQQJBgcGCQYHCAcIBwgHCgUK
BQwFCgMMAQwBDgEMAA4BO+YCnwwJEQkRCQ8JDwsNCQ0LDQkLCwsJCQsLCQkLB
wsHCwcLBwsFCwcNAwsFDQMLBQ0BDQMNAQ0DDQENAQ0ADQENAA0AVwAbDQDSA
hoPQgAIAAgABgAIAgYCCAIGAgYEBgQGBAQEBAQEBgQEBAYCBgC4CRES6gIAEREAo
wo=\",\n
\"k\":\"BgABARKoA/QFIAC0AYoD5gIAjwK5BJICwwTfAgDDAbIDFwAAnwMSEeUCABISAJIL
ERLmAgAREQCvBQ==\",\n
\"n\":\"BgABAW1yggmQAU8GBAgEBgQGBgYCCAQGBAYEBgQIAgYECAQGAggEBgIIBAg
CCAQIAggCCAIIAgoACAIKAAgCCgAKAgoADAAKAgwAFgAWARQAFAEUAxQDFAMSAxIF
EgUQBRIHEAkOBxAJDgsOCwwLDA0MDQoPCA8IEQgRBhEGEwYVBBUEFQIXAhkCGQD
tBRQR5QIAFBAA/AUACAEIAQYBCAMGBQQFBgUEBwQFBAcCBwIHAgcCCQIHAAcACQ
AHAQcABwMHAQUDBwMFAwUFBQUDBQEFAwcBBwAHAPkFEhHjAgASEgDwCBAA\",\n
\"m\":\"BgABAZoBfoIJigFbDAwMCg4KDggOCA4IDgYQBhAGEAQQBBAEEAISAhACEgAm
ASQDJAciCyANHhEcFRwXDg4QDBAKEAwQCBAKEggSBhIGEgYSBBQEEgIUAhICFAAU
ABQBEgEUARIDEgMSAxIFEgUQBxAHEAcQBw4JDgkOCw4LDAsMDQoNCg8KDwgPCBEI
EQYRBBMEEwQTAhMCFQAVAP0FEhHlAgASEgCCBgAIAQgBBgEGAwYFBgUEBQQHBA
UEBwIHAgcCBwIJAAcABwAJAAcBBwEHAQUBBwMFAwUDBQMDBQMFAwUBBQEHAQc
AgQYSEeUCABISAIIGAAgBCAEGAQYDBgUGBQQFBAcEBQQHAgcCBwIHAgkABwAHAA
kABwEHAQcBBQEHAwUDBQMFAwMFAwUDBQEFAQcBBwCBBhIR5QIAEhIA8AgYAA==\"
,\n \"l\":\"BgABAQnAAwDrAgASFgDWCxEa6gIAERkA0wsUFw==\",\n
\"y\":\"BgABAZ8BogeNAg8ZERkRFxEVExMVERUPFQ8XDRcLGQkZBxsFGwUdAR0BDQA
LAA0ADQINAAsCDQANAg0CDQILAg0EDQINBA0GDQQNBg0EDQYNCA0GDwgNCA0IDQ
gPCg0KDwwNDA8MDw4PDqIB7gEQDRALEAkQCQ4JEAcOBw4FDgUOAwwFDgMMAQw
BDAEMAQwACgEKAAoACAIIAAgCCAIGAggCBgIGBAYCBgQEAgYEAqIBAQADAAEBAwA
DAAMABQADAAUAAwAFAAMABQAFAAMABQA3ABMAEwIRABECEwQRAg8EEQQPBBE
GDwgNCA8IDQgNCg0MDQwLDAkOCw4JDgcQBxAHEgUSBRQFFAMWARgDGAEaABwA
9AUTEuQCABEPAP8FAAUCBQAFAgUEBQIDBAUEAwQDBgMEAQYDBgEGAAgBBgCAA
QAAvAYREuICABMPAP0K\",\n
\"q\":\"BgABAmj0A4YJFgAWARQAEgESAxADEAMOAw4FDgUMBQ4HDgcOBwwJDgmeAU
4A2QwWGesCABYaAN4DAwADAAMBAwADAAUAAwADAAMABQAFAAUABwAHAQcACQ

AVABUCFQATAhUCEwQRAhMEEQQRBhEGDwgPCA8IDQoNDA0MCwwLDgkOCRAJEAk
QBxIHEgUUBRYDFgMYARoBGgAcAP4CABYCFgIWBBYEFAQSBhQIEggSCBAKEgoQDA
4MDgwODg4ODBAMDgwQChIIEAoSCBIGEgYUBhQEFAQWAhYCFgIWAApbkQYSKy4Re
AAAjARTEjkRHykJMwDvAg==\",\n
\"p\":\"BgABAmiCBIYJFgAWARYBFAEWAxQDEgUUBRIFEgcSBxAJEAkQCQ4LDgsOCww
NDA0KDwoPCg8IEQgRCBEGEwQTBhMCFQQVAhUAFQD9AgAbARkBFwMXAxcDEwUTB
xMHEQcRCQ8JDQsNCw0LCw0LDQkPCQ0JDwURBxEFEQURAxMDEQMTARUBEwEVAR
UBFQAJAAcABwAFAAcABQAFAAMAAwADAAUAAwIDAAMAAwIDAADdAxYZ6wIAFhoA2g
yeAU0OCgwIDgoMCA4GDgYMBg4GDgQQBBAEEgQUAhQCFgIWAApcoQMJNB8qNxJVE
QCLBHgALhISLADwAg==\",\n
\"o\":\"BgABAoMB8gOICRYAFgEWARQBFgMUAxIDFAUSBRIHEgcQBxAJEAkOCw4LDgsM
DQwNCg8KDwoPCg8IEQgRBhMGEwQTBBMCFQIVABcAiwMAFwEVARUDEwMTAxMFEw
cRBxEHDwkPCQ8LDQsNCw0NCw0LDwkNCw8HEQkPBxEHEQcRBRMFEwMTAxUDFQE
VABUAFQAVAhUCFQITBBMEEwYTBhEGEQgRCA8KDwoPCg0KDQwNDAsOCw4JDgkQC
RAJEgcSBxIFFAUUAxQDFgEWARYAFgCMAwAYAhYCFgQUBBQEFAYUCBIIEggQChAKE
AwODA4MDg4MDgwQCg4KEgoQChIIEggSBhQGEgYUBBYEFAIWAhYCFgALYv0CHTZBF
EMRHTcAjwMcNUITQhIiOACQAw==\",\n
\"r\":\"BgACAQRigAkQAA8AAAABShAAhAFXDAwODAwKDgoOCBAIDgYQBhAEEAQQBBA
EEAISABACEAAQAA4BEAAQARADEAEQAxADEAUSBRIHFAcUCxQLFA0WDVJFsQHzA
QsMDQwLCgkICwgLCAkGCQYJBAkGBwIJBAcCBwQHAAcCBwAFAgcABQAHAQUABQEF
AQUBBQEDAQUBAwMDAQMDAwEAmwYSEeMCABISAO4IEAA=\",\n
\"u\":\"BgABAV2KBwGPAVANCQsHDQcNBw0FCwUNBQ0FDQMPAw8DEQMTARMBFQEV
ABUAFQITABMEEwITBBMEEQQRBhEGDwYRCA8KDQgPCg0MDQwLDAsOCRALDgcQB
xIHEgUUBRQFFAMWAxgBGAEYARoA7gUTEuYCABMPAPsFAAcCBwIFBAcCBQYDBgUG
AwgDBgMIAQgBCAEIAQoBCAAIAAoACAIIAggCCAIGBAgEBgQGBgYGBAYCBgQIAggAC
AD6BRES5AIAEREA7wgPAA==\",\n
\"s\":\"BgABAasC/gLwBQoDCgMMBQ4DDgUOBRAFEAUSBRAHEgcQCRIJEAkSCxALEAs
QDRANDg0ODw4PDA8MDwoRChEIEwYTBBcCFQIXABkBGQEXAxcFFQUTBRMHEwcRC
REJDwkNCQ8LDQ0LCwsNCw0JDQkPBw8HDwUPBREDEQMRAREDEQETABEBEwARA
BMADwIRABECEQIRBBMCEwQVBBUEFQYVBhMIFwgVChUKFQxgsAIIAwYDCAMKAQgD
CAMKAQoDCgEKAwoBCgMKAQwDCgEKAwoBDAMKAQoBCgEMAQoACgEKAAoBCgAK
AQgACgAIAQgABgoECAIKAgoCCgAMAQoBDAUEBwIHBAcEBwIHBAkECQQJBAkECQYL
BAkGCwYJBgsGCwYJCAsGCwgJBgsICQgLCAkICwgJCgkKCQoJCgcKCQwHDAcMBwwF
DAcMAw4FDAMOAw4BDgMQARAAEAESABIAEgIQAg4CDgIOBA4CDgQMBAwEDAQMBg
oECgYKBgoGCgYIBggGCAgIBggGBgYIBgYGBgYGBgYGBAgGBgQIBAYECAQQChIIEggS
BhIEEgQSBBQCFAISABQAEgASABIAEgESARIBEAEQAxIDDgMQAxADDgUOBQwDDAM
MAwoDCAMIAQYBe6cCAwIDAgUAAwIFAgUCBwIFAgcCBQIHAgUCBwIHAAUCBwIHAgUA
BwIHAgcABQIHAAcCBwAFAgUABQIFAAUABQIDAAEAAQABAQEAAQEBAQEBAQEBAQ
EDAQEAAwEBAQMAAwEDAAMBAwADAQMAAwABAQMAAwADAAEAAwIBAAMCAQQDA
gE=\",\n
\"t\":\"BgABAUe8BLACWAAaEADRAhsOaQANAA0ADwINAA0CDQANAg0CDQINBA0CCw
YNBA0GCwYNBgsIDQgLCAsKCwgJDAsKCQwJDAkOCQ4HEAcSBxIHEgUUAOAEawAVE
QDWAhYTbAAAygIVFOYCABUXAMUCogEAFhQA1QIVEqEBAADzAwIFBAMEBQQDBAM
EAwYDBgMGAwYBCAEGAQgBBgEIAAgA\",\n
\"w\":\"BgABARz8BsAEINYCKNgBERLuAgARD+8B3QgSEc0CABQSW7YCV7UCFBHJAgA
SEpMC3AgREvACABERmAHxBDDaAVeYAxES7gIAEREo1QE81wIIAA==\",\n
\"z\":\"BgABAQ6cA9AGuQIAFw8AzAIaC9QFAAAr9wKjBuACABYQAMsCGQyZBgCaA9AG\"
\n }';\nBEGIN\n\n IF font IS NULL THEN\n   font := font_default;\n END IF;\n\n -- For

character spacing, use m as guide size\n geom := ST_GeomFromTWKB(decode(font->>'m', 'base64'));\n m_width := ST_XMax(geom) - ST_XMin(geom);\n spacing := m_width / 12;\n\n letterarray := regexp_split_to_array(replace(letters, ' ', E'\\t'), E'');\n FOREACH letter IN ARRAY letterarray\n LOOP\n   geom := ST_GeomFromTWKB(decode(font->>(letter), 'base64'));\n   -- Chars are not already zeroed out, so do it now\n   geom := ST_Translate(geom, -1 * ST_XMin(geom), 0.0);\n   -- unknown characters are treated as spaces\n   IF geom IS NULL THEN\n   -- spaces are a \"quarter m\" in width\n   width := m_width / 3.5;\n   ELSE\n   width := (ST_XMax(geom) - ST_XMin(geom));\n   END IF;\n   geom := ST_Translate(geom, position, 0.0);\n   -- Tighten up spacing when characters have a large gap\n   -- between them like Yo or To\n   adjustment := 0.0;\n   IF prevgeom IS NOT NULL AND geom IS NOT NULL THEN\n   dist = ST_Distance(prevgeom, geom);\n   IF dist > spacing THEN\n   adjustment = spacing - dist;\n   geom := ST_Translate(geom, adjustment, 0.0);\n   END IF;\n   END IF;\n   prevgeom := geom;\n   position := position + width + spacing + adjustment;\n   wordarr := array_append(wordarr, geom);\n END LOOP;\n -- apply the start point and scaling options\n wordgeom := ST_CollectionExtract(ST_Collect(wordarr));\n wordgeom := ST_Scale(wordgeom,\n text_height/font_default_height,\n   text_height/font_default_height);\n return wordgeom;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linecrossingdirection",
    "arguments": "line1 geometry, line2 geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linecrossingdirection(line1 geometry, line2 geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3', $function$ST_LineCrossingDirection$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefromencodedpolyline",
    "arguments": "txtin text, nprecision integer DEFAULT 5",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefromencodedpolyline(txtin text, nprecision integer DEFAULT 5)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$line_from_encoded_polyline$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefrommultipoint",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefrommultipoint(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_line_from_mpoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefromtext(text, integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$\n\tSELECT CASE WHEN public.geometrytype(public.ST_GeomFromText($1, $2)) = 'LINESTRING'\n\tTHEN public.ST_GeomFromText($1,$2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefromtext(text)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS $function$\n\tSELECT CASE WHEN public.geometrytype(public.ST_GeomFromText($1)) = 'LINESTRING'\n\tTHEN public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefromwkb(bytea, integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$\n\tSELECT CASE WHEN public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'LINESTRING'\n\tTHEN public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linefromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linefromwkb(bytea)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$\n\tSELECT CASE WHEN
```

```
public.geometrytype(public.ST_GeomFromWKB($1)) = 'LINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_lineinterpolatepoint",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_lineinterpolatepoint(geometry,
double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_line_interpolate_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_lineinterpolatepoints",
    "arguments": "geometry, double precision, repeat boolean DEFAULT true",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_lineinterpolatepoints(geometry,
double precision, repeat boolean DEFAULT true)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_line_interpolate_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linelocatepoint",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linelocatepoint(geom1
geometry, geom2 geometry)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_line_locate_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linemerge",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linemerge(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$linemerge$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_linemerge",
    "arguments": "geometry, boolean",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linemerge(geometry,
boolean)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS '$libdir/postgis-3', $function$linemerge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linestringfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linestringfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'LINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linestringfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linestringfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'LINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linesubstring",
    "arguments": "geometry, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linesubstring(geometry, double
precision, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_line_substring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_linetocurve",
    "arguments": "geometry geometry",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_linetocurve(geometry
geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_line_desegmentize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_locatealong",
    "arguments": "geometry geometry, measure double precision, leftrightoffset double
precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_locatealong(geometry
geometry, measure double precision, leftrightoffset double precision DEFAULT 0.0)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$ST_LocateAlong$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_locatebetween",
    "arguments": "geometry geometry, frommeasure double precision, tomeasure double
precision, leftrightoffset double precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_locatebetween(geometry
geometry, frommeasure double precision, tomeasure double precision, leftrightoffset double
precision DEFAULT 0.0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_LocateBetween$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_locatebetweenelevations",
    "arguments": "geometry geometry, fromelevation double precision, toelevation double
precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_locatebetweenelevations(geometry geometry, fromelevation double precision,
toelevation double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$ST_LocateBetweenElevations$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_longestline",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_longestline(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL
SAFE STRICT COST 10000\nAS $function$SELECT
public._ST_LongestLine(public.ST_ConvexHull($1), public.ST_ConvexHull($2))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_m",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_m(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_m_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makebox2d",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "box2d",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makebox2d(geom1 geometry,
geom2 geometry)\n RETURNS box2d\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$BOX2D_construct$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makeenvelope",
    "arguments": "double precision, double precision, double precision, double precision,
integer DEFAULT 0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makeenvelope(double
precision, double precision, double precision, double precision, integer DEFAULT 0)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_MakeEnvelope$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makeline",
    "arguments": "geometry[]",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makeline(geometry[])\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_makeline_garray$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "st_makeline",
   "arguments": "geom1 geometry, geom2 geometry",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_makeline(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makeline$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_makepoint",
   "arguments": "double precision, double precision",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_makepoint(double precision,
double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makepoint$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_makepoint",
   "arguments": "double precision, double precision, double precision",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_makepoint(double precision,
double precision, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_makepoint$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_makepoint",
   "arguments": "double precision, double precision, double precision, double precision",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_makepoint(double precision,
double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE
c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_makepoint$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_makepointm",
   "arguments": "double precision, double precision, double precision",
   "return_type": "geometry",
   "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_makepointm(double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makepoint3dm$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makepolygon",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makepolygon(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makepoly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makepolygon",
    "arguments": "geometry, geometry[]",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makepolygon(geometry, geometry[])\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makepoly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makevalid",
    "arguments": "geom geometry, params text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makevalid(geom geometry, params text)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_MakeValid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_makevalid",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_makevalid(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_MakeValid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_maxdistance",
```

```
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_maxdistance(geom1 geometry,
geom2 geometry)\n RETURNS double precision\n LANGUAGE sql\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS $function$SELECT
public._ST_MaxDistance(public.ST_ConvexHull($1),
public.ST_ConvexHull($2))$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_maximuminscribedcircle",
    "arguments": "geometry, OUT center geometry, OUT nearest geometry, OUT radius
double precision",
    "return_type": "record",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_maximuminscribedcircle(geometry, OUT center geometry, OUT nearest geometry,
OUT radius double precision)\n RETURNS record\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_MaximumInscribedCircle$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_memsize",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_memsize(geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_mem_size$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_minimumboundingcircle",
    "arguments": "inputgeom geometry, segs_per_quarter integer DEFAULT 48",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_minimumboundingcircle(inputgeom geometry, segs_per_quarter integer DEFAULT
48)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS '$libdir/postgis-3', $function$ST_MinimumBoundingCircle$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_minimumboundingradius",
    "arguments": "geometry, OUT center geometry, OUT radius double precision",
    "return_type": "record",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_minimumboundingradius(geometry, OUT center geometry, OUT radius double
precision)\n RETURNS record\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS '$libdir/postgis-3', $function$ST_MinimumBoundingRadius$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_minimumclearance",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_minimumclearance(geometry)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_MinimumClearance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_minimumclearanceline",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_minimumclearanceline(geometry)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_MinimumClearanceLine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mlinefromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mlinefromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'MULTILINESTRING'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mlinefromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mlinefromtext(text, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
```

500\nAS $function$\n\tSELECT CASE\n\tWHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) = 'MULTILINESTRING'\n\tTHEN
public.ST_GeomFromText($1,$2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mlinefromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mlinefromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'MULTILINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mlinefromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mlinefromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'MULTILINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpointfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpointfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'MULTIPOINT'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpointfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpointfromtext(text, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST

```
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) = 'MULTIPOINT'\n\tTHEN
ST_GeomFromText($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpointfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpointfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'MULTIPOINT'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpointfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpointfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'MULTIPOINT'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpolyfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpolyfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'MULTIPOLYGON'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpolyfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpolyfromtext(text, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
```

```
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) = 'MULTIPOLYGON'\n\tTHEN
public.ST_GeomFromText($1,$2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpolyfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpolyfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT\nAS
$function$\n\tSELECT CASE WHEN public.geometrytype(public.ST_GeomFromWKB($1)) =
'MULTIPOLYGON'\n\tTHEN public.ST_GeomFromWKB($1)\n\tELSE NULL
END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_mpolyfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_mpolyfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'MULTIPOLYGON'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multi",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multi(geometry)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_force_multi$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multilinefromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multilinefromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
```

```
public.geometrytype(public.ST_GeomFromWKB($1)) = 'MULTILINESTRING'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multilinestringfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multilinestringfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$SELECT public.ST_MLineFromText($1)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multilinestringfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multilinestringfromtext(text,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 500\nAS $function$SELECT public.ST_MLineFromText($1, $2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipointfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipointfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$SELECT public.ST_MPointFromText($1)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipointfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipointfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'MULTIPOINT'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipointfromwkb",
```

```
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipointfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1,$2)) = 'MULTIPOINT'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipolyfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipolyfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'MULTIPOLYGON'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipolyfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipolyfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'MULTIPOLYGON'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipolygonfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipolygonfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$SELECT public.ST_MPolyFromText($1)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_multipolygonfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_multipolygonfromtext(text,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 500\nAS $function$SELECT public.ST_MPolyFromText($1, $2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ndims",
    "arguments": "geometry",
    "return_type": "smallint",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ndims(geometry)\n RETURNS
smallint\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/postgis-3',
$function$LWGEOM_ndims$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_node",
    "arguments": "g geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_node(g geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_Node$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_normalize",
    "arguments": "geom geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_normalize(geom geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$ST_Normalize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_npoints",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_npoints(geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_npoints$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_nrings",
```

```
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_nrings(geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_nrings$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_numgeometries",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_numgeometries(geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_numgeometries_collection$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_numinteriorring",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_numinteriorring(geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_numinteriorrings_polygon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_numinteriorrings",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_numinteriorrings(geometry)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_numinteriorrings_polygon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_numpatches",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_numpatches(geometry)\n RETURNS integer\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$\n\tSELECT CASE WHEN public.ST_GeometryType($1) = 'ST_PolyhedralSurface'\n\tTHEN public.ST_NumGeometries($1)\n\tELSE NULL END\n\t$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_numpoints",
    "arguments": "geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_numpoints(geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_numpoints_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_offsetcurve",
    "arguments": "line geometry, distance double precision, params text DEFAULT ''::text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_offsetcurve(line geometry,
distance double precision, params text DEFAULT ''::text)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$ST_OffsetCurve$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_orderingequals",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_orderingequals(geom1
geometry, geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS
'$libdir/postgis-3', $function$LWGEOM_same$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_orientedenvelope",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_orientedenvelope(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_OrientedEnvelope$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_overlaps",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_overlaps(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$overlaps$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_patchn",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_patchn(geometry, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN public.ST_GeometryType($1) =
'ST_PolyhedralSurface'\n\tTHEN public.ST_GeometryN($1, $2)\n\tELSE NULL
END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_perimeter",
    "arguments": "geog geography, use_spheroid boolean DEFAULT true",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_perimeter(geog geography,
use_spheroid boolean DEFAULT true)\n RETURNS double precision\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_perimeter$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_perimeter",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_perimeter(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_perimeter2d_poly$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_perimeter2d",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_perimeter2d(geometry)\n
RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_perimeter2d_poly$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_point",
    "arguments": "double precision, double precision, srid integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_point(double precision, double
precision, srid integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_Point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_point",
    "arguments": "double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_point(double precision, double
precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_makepoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointfromgeohash",
    "arguments": "text, integer DEFAULT NULL::integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointfromgeohash(text, integer
DEFAULT NULL::integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE COST 50\nAS '$libdir/postgis-3',
$function$point_from_geohash$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'POINT'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointfromtext",
    "arguments": "text, integer",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointfromtext(text, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) = 'POINT'\n\tTHEN
public.ST_GeomFromText($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'POINT'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointfromwkb(bytea, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'POINT'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointinsidecircle",
    "arguments": "geometry, double precision, double precision, double precision",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointinsidecircle(geometry,
double precision, double precision, double precision)\n RETURNS boolean\n LANGUAGE
c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$LWGEOM_inside_circle_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointm",
    "arguments": "xcoordinate double precision, ycoordinate double precision, mcoordinate
double precision, srid integer DEFAULT 0",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointm(xcoordinate double
precision, ycoordinate double precision, mcoordinate double precision, srid integer
DEFAULT 0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_PointM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointn",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointn(geometry, integer)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_pointn_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointonsurface",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointonsurface(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$pointonsurface$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_points",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_points(geometry)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$ST_Points$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_pointz",
    "arguments": "xcoordinate double precision, ycoordinate double precision, zcoordinate
double precision, srid integer DEFAULT 0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointz(xcoordinate double
precision, ycoordinate double precision, zcoordinate double precision, srid integer DEFAULT
0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_PointZ$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_pointzm",
    "arguments": "xcoordinate double precision, ycoordinate double precision, zcoordinate
double precision, mcoordinate double precision, srid integer DEFAULT 0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_pointzm(xcoordinate double
precision, ycoordinate double precision, zcoordinate double precision, mcoordinate double
precision, srid integer DEFAULT 0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_PointZM$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polyfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polyfromtext(text, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1, $2)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromText($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polyfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polyfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromText($1)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromText($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polyfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polyfromwkb(bytea, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
```

```
public.geometrytype(public.ST_GeomFromWKB($1, $2)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polyfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polyfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polygon",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygon(geometry, integer)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT public.ST_SetSRID(public.ST_MakePolygon($1),
$2)\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polygonfromtext",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygonfromtext(text)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS $function$SELECT public.ST_PolyFromText($1)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polygonfromtext",
    "arguments": "text, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygonfromtext(text,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 500\nAS $function$SELECT public.ST_PolyFromText($1, $2)$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_polygonfromwkb",
    "arguments": "bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygonfromwkb(bytea)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromWKB($1)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polygonfromwkb",
    "arguments": "bytea, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygonfromwkb(bytea,
integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS $function$\n\tSELECT CASE WHEN
public.geometrytype(public.ST_GeomFromWKB($1,$2)) = 'POLYGON'\n\tTHEN
public.ST_GeomFromWKB($1, $2)\n\tELSE NULL END\n\t$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_polygonize",
    "arguments": "geometry[]",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_polygonize(geometry[])\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$polygonize_garray$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_project",
    "arguments": "geog geography, distance double precision, azimuth double precision",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_project(geog geography,
distance double precision, azimuth double precision)\n RETURNS geography\n LANGUAGE
c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$geography_project$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_quantizecoordinates",
    "arguments": "g geometry, prec_x integer, prec_y integer DEFAULT NULL::integer, prec_z
integer DEFAULT NULL::integer, prec_m integer DEFAULT NULL::integer",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_quantizecoordinates(g
geometry, prec_x integer, prec_y integer DEFAULT NULL::integer, prec_z integer DEFAULT
NULL::integer, prec_m integer DEFAULT NULL::integer)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE COST 500\nAS '$libdir/postgis-3',
$function$ST_QuantizeCoordinates$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_reduceprecision",
    "arguments": "geom geometry, gridsize double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_reduceprecision(geom
geometry, gridsize double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$ST_ReducePrecision$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_relate",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_relate(geom1 geometry,
geom2 geometry)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$relate_full$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_relate",
    "arguments": "geom1 geometry, geom2 geometry, text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_relate(geom1 geometry,
geom2 geometry, text)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$relate_pattern$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_relate",
    "arguments": "geom1 geometry, geom2 geometry, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_relate(geom1 geometry,
geom2 geometry, integer)\n RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$relate_full$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_relatematch",
    "arguments": "text, text",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_relatematch(text, text)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$ST_RelateMatch$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_removepoint",
    "arguments": "geometry, integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_removepoint(geometry, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_removepoint$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_removerepeatedpoints",
    "arguments": "geom geometry, tolerance double precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_removerepeatedpoints(geom geometry, tolerance double precision DEFAULT 0.0)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_RemoveRepeatedPoints$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_reverse",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_reverse(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_reverse$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotate",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotate(geometry, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1,  cos($2), -sin($2), 0,  sin($2), cos($2), 0,  0, 0, 1,  0, 0, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotate",
    "arguments": "geometry, double precision, geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotate(geometry, double precision, geometry)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1,  cos($2), -sin($2), 0,  sin($2),  cos($2), 0, 0, 0, 1, public.ST_X($3) - cos($2) * public.ST_X($3) + sin($2) * public.ST_Y($3), public.ST_Y($3) - sin($2) * public.ST_X($3) - cos($2) * public.ST_Y($3), 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotate",
    "arguments": "geometry, double precision, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotate(geometry, double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1,  cos($2), -sin($2), 0,  sin($2),  cos($2), 0, 0, 0, 1,\t$3 - cos($2) * $3 + sin($2) * $4, $4 - sin($2) * $3 - cos($2) * $4, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotatex",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotatex(geometry, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1, 1, 0, 0, 0, cos($2), -sin($2), 0, sin($2), cos($2), 0, 0, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotatey",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotatey(geometry, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1,  cos($2), 0, sin($2),  0, 1, 0, -sin($2), 0, cos($2), 0,  0, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_rotatez",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_rotatez(geometry, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Rotate($1, $2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_scale",
    "arguments": "geometry, geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_scale(geometry, geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_Scale$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_scale",
    "arguments": "geometry, geometry, origin geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_scale(geometry, geometry, origin geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_Scale$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_scale",
    "arguments": "geometry, double precision, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_scale(geometry, double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Scale($1, public.ST_MakePoint($2, $3, $4))$function$\n"
  },
  {
    "schema": "public",
```

```
    "function_name": "st_scale",
    "arguments": "geometry, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_scale(geometry, double
precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Scale($1, $2, $3,
1)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_scroll",
    "arguments": "geometry, geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_scroll(geometry, geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$ST_Scroll$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_segmentize",
    "arguments": "geog geography, max_segment_length double precision",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_segmentize(geog geography,
max_segment_length double precision)\n RETURNS geography\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3',
$function$geography_segmentize$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_segmentize",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_segmentize(geometry, double
precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_segmentize2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_seteffectivearea",
    "arguments": "geometry, double precision DEFAULT '-1'::integer, integer DEFAULT 1",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_seteffectivearea(geometry,
double precision DEFAULT '-1'::integer, integer DEFAULT 1)\n RETURNS geometry\n
```

```
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$LWGEOM_SetEffectiveArea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_setpoint",
    "arguments": "geometry, integer, geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_setpoint(geometry, integer,
geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_setpoint_linestring$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_setsrid",
    "arguments": "geog geography, srid integer",
    "return_type": "geography",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_setsrid(geog geography, srid
integer)\n RETURNS geography\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_set_srid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_setsrid",
    "arguments": "geom geometry, srid integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_setsrid(geom geometry, srid
integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/postgis-3', $function$LWGEOM_set_srid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_sharedpaths",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_sharedpaths(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_SharedPaths$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_shiftlongitude",
    "arguments": "geometry",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_shiftlongitude(geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_longitude_shift$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_shortestline",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_shortestline(geom1 geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS '$libdir/postgis-3', $function$LWGEOM_shortestline2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_simplify",
    "arguments": "geometry, double precision, boolean",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_simplify(geometry, double precision, boolean)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_simplify2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_simplify",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_simplify(geometry, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$LWGEOM_simplify2d$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_simplifypolygonhull",
    "arguments": "geom geometry, vertex_fraction double precision, is_outer boolean DEFAULT true",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_simplifypolygonhull(geom geometry, vertex_fraction double precision, is_outer boolean DEFAULT true)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_SimplifyPolygonHull$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_simplifypreservetopology",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.st_simplifypreservetopology(geometry, double precision)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS
'$libdir/postgis-3', $function$topologypreservesimplify$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_simplifyvw",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_simplifyvw(geometry, double
precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3',
$function$LWGEOM_SetEffectiveArea$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_snap",
    "arguments": "geom1 geometry, geom2 geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_snap(geom1 geometry, geom2
geometry, double precision)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE
PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_Snap$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_snaptogrid",
    "arguments": "geometry, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_snaptogrid(geometry, double
precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE
PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_SnapToGrid($1, 0,
0, $2, $3)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_snaptogrid",
```

```
    "arguments": "geom1 geometry, geom2 geometry, double precision, double precision,
double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_snaptogrid(geom1 geometry,
geom2 geometry, double precision, double precision, double precision, double precision)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_snaptogrid_pointoff$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_snaptogrid",
    "arguments": "geometry, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_snaptogrid(geometry, double
precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE
STRICT COST 50\nAS $function$SELECT public.ST_SnapToGrid($1, 0, 0, $2,
$2)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_snaptogrid",
    "arguments": "geometry, double precision, double precision, double precision, double
precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_snaptogrid(geometry, double
precision, double precision, double precision, double precision)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$LWGEOM_snaptogrid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_split",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_split(geom1 geometry, geom2
geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_Split$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_square",
    "arguments": "size double precision, cell_i integer, cell_j integer, origin geometry
DEFAULT '0101000000000000000000000000000000000000'::geometry",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_square(size double precision,
cell_i integer, cell_j integer, origin geometry DEFAULT
'0101000000000000000000000000000000000000'::geometry)\n RETURNS geometry\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_Square$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_squaregrid",
    "arguments": "size double precision, bounds geometry, OUT geom geometry, OUT i
integer, OUT j integer",
    "return_type": "SETOF record",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_squaregrid(size double
precision, bounds geometry, OUT geom geometry, OUT i integer, OUT j integer)\n
RETURNS SETOF record\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
500\nAS '$libdir/postgis-3', $function$ST_ShapeGrid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_srid",
    "arguments": "geom geometry",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_srid(geom geometry)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_get_srid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_srid",
    "arguments": "geog geography",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_srid(geog geography)\n
RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_get_srid$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_startpoint",
    "arguments": "geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_startpoint(geometry)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_startpoint_linestring$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_subdivide",
    "arguments": "geom geometry, maxvertices integer DEFAULT 256, gridsize double
precision DEFAULT '-1.0'::numeric",
    "return_type": "SETOF geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_subdivide(geom geometry,
maxvertices integer DEFAULT 256, gridsize double precision DEFAULT '-1.0'::numeric)\n
RETURNS SETOF geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT
COST 10000\nAS '$libdir/postgis-3', $function$ST_Subdivide$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_summary",
    "arguments": "geography",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_summary(geography)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_summary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_summary",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_summary(geometry)\n
RETURNS text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_summary$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_swapordinates",
    "arguments": "geom geometry, ords cstring",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_swapordinates(geom
geometry, ords cstring)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL
SAFE STRICT COST 50\nAS '$libdir/postgis-3', $function$ST_SwapOrdinates$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_symdifference",
    "arguments": "geom1 geometry, geom2 geometry, gridsize double precision DEFAULT
'-1.0'::numeric",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_symdifference(geom1
geometry, geom2 geometry, gridsize double precision DEFAULT '-1.0'::numeric)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$ST_SymDifference$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_symmetricdifference",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_symmetricdifference(geom1
geometry, geom2 geometry)\n RETURNS geometry\n LANGUAGE sql\nAS
$function$SELECT ST_SymDifference(geom1, geom2, -1.0);$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_tileenvelope",
    "arguments": "zoom integer, x integer, y integer, bounds geometry DEFAULT
'0102000020110F00000200000093107C45F81B73C193107C45F81B73C193107C45F81B7
34193107C45F81B7341'::geometry, margin double precision DEFAULT 0.0",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_tileenvelope(zoom integer, x
integer, y integer, bounds geometry DEFAULT
'0102000020110F00000200000093107C45F81B73C193107C45F81B73C193107C45F81B7
34193107C45F81B7341'::geometry, margin double precision DEFAULT 0.0)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$ST_TileEnvelope$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_touches",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_touches(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$touches$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_transform",
    "arguments": "geometry, integer",
    "return_type": "geometry",
```

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_transform(geometry, integer)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$transform$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_transform",
    "arguments": "geom geometry, to_proj text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_transform(geom geometry, to_proj text)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS $function$SELECT public.postgis_transform_geometry($1, proj4text, $2, 0)\n\tFROM spatial_ref_sys WHERE srid=public.ST_SRID($1);$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_transform",
    "arguments": "geom geometry, from_proj text, to_proj text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_transform(geom geometry, from_proj text, to_proj text)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS $function$SELECT public.postgis_transform_geometry($1, $2, $3, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_transform",
    "arguments": "geom geometry, from_proj text, to_srid integer",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_transform(geom geometry, from_proj text, to_srid integer)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS $function$SELECT public.postgis_transform_geometry($1, $2, proj4text, $3)\n\tFROM spatial_ref_sys WHERE srid=$3;$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_translate",
    "arguments": "geometry, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_translate(geometry, double precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE
```

PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Translate($1, $2, $3, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_translate",
    "arguments": "geometry, double precision, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_translate(geometry, double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1, 1, 0, 0, 0, 1, 0, 0, 0, 1, $2, $3, $4)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_transscale",
    "arguments": "geometry, double precision, double precision, double precision, double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_transscale(geometry, double precision, double precision, double precision, double precision)\n RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS $function$SELECT public.ST_Affine($1, $4, 0, 0, 0, $5, 0,\n\t\t0, 0, 1, $2 * $4, $3 * $5, 0)$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_triangulatepolygon",
    "arguments": "g1 geometry",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_triangulatepolygon(g1 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_TriangulatePolygon$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_unaryunion",
    "arguments": "geometry, gridsize double precision DEFAULT '-1.0'::numeric",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_unaryunion(geometry, gridsize double precision DEFAULT '-1.0'::numeric)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_UnaryUnion$function$\n"
  },

```
 {
   "schema": "public",
   "function_name": "st_union",
   "arguments": "geom1 geometry, geom2 geometry",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_union(geom1 geometry,
geom2 geometry)\n RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000\nAS '$libdir/postgis-3', $function$ST_Union$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_union",
   "arguments": "geometry[]",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_union(geometry[])\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
10000\nAS '$libdir/postgis-3', $function$pgis_union_geometry_array$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_union",
   "arguments": "geom1 geometry, geom2 geometry, gridsize double precision",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_union(geom1 geometry,
geom2 geometry, gridsize double precision)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 10000\nAS '$libdir/postgis-3',
$function$ST_Union$function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_voronoilines",
   "arguments": "g1 geometry, tolerance double precision DEFAULT 0.0, extend_to geometry
DEFAULT NULL::geometry",
   "return_type": "geometry",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.st_voronoilines(g1 geometry,
tolerance double precision DEFAULT 0.0, extend_to geometry DEFAULT NULL::geometry)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$
SELECT public._ST_Voronoi(g1, extend_to, tolerance, false) $function$\n"
 },
 {
   "schema": "public",
   "function_name": "st_voronoipolygons",
   "arguments": "g1 geometry, tolerance double precision DEFAULT 0.0, extend_to geometry
DEFAULT NULL::geometry",
```

```
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_voronoipolygons(g1 geometry,
tolerance double precision DEFAULT 0.0, extend_to geometry DEFAULT NULL::geometry)\n
RETURNS geometry\n LANGUAGE sql\n IMMUTABLE PARALLEL SAFE\nAS $function$
SELECT public._ST_Voronoi(g1, extend_to, tolerance, true) $function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_within",
    "arguments": "geom1 geometry, geom2 geometry",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_within(geom1 geometry,
geom2 geometry)\n RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT COST 10000 SUPPORT postgis_index_supportfn\nAS '$libdir/postgis-3',
$function$within$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_wkbtosql",
    "arguments": "wkb bytea",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_wkbtosql(wkb bytea)\n
RETURNS geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST
50\nAS '$libdir/postgis-3', $function$LWGEOM_from_WKB$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_wkttosql",
    "arguments": "text",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_wkttosql(text)\n RETURNS
geometry\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 500\nAS
'$libdir/postgis-3', $function$LWGEOM_from_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_wrapx",
    "arguments": "geom geometry, wrap double precision, move double precision",
    "return_type": "geometry",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_wrapx(geom geometry, wrap
double precision, move double precision)\n RETURNS geometry\n LANGUAGE c\n
IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS '$libdir/postgis-3',
$function$ST_WrapX$function$\n"
```

```
  },
  {
    "schema": "public",
    "function_name": "st_x",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_x(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_x_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_xmax",
    "arguments": "box3d",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_xmax(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_xmax$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_xmin",
    "arguments": "box3d",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_xmin(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_xmin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_y",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_y(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_y_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ymax",
    "arguments": "box3d",
    "return_type": "double precision",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.st_ymax(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_ymax$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_ymin",
    "arguments": "box3d",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_ymin(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_ymin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_z",
    "arguments": "geometry",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_z(geometry)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_z_point$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_zmax",
    "arguments": "box3d",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_zmax(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_zmax$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_zmflag",
    "arguments": "geometry",
    "return_type": "smallint",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_zmflag(geometry)\n
RETURNS smallint\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$LWGEOM_zmflag$function$\n"
  },
  {
    "schema": "public",
    "function_name": "st_zmin",
    "arguments": "box3d",
```

```
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.st_zmin(box3d)\n RETURNS
double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/postgis-3', $function$BOX3D_zmin$function$\n"
  },
  {
    "schema": "public",
    "function_name": "subvector",
    "arguments": "halfvec, integer, integer",
    "return_type": "halfvec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.subvector(halfvec, integer,
integer)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE
STRICT\nAS '$libdir/vector', $function$halfvec_subvector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "subvector",
    "arguments": "vector, integer, integer",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.subvector(vector, integer,
integer)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$subvector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "text",
    "arguments": "geometry",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.text(geometry)\n RETURNS
text\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT COST 50\nAS
'$libdir/postgis-3', $function$LWGEOM_to_text$function$\n"
  },
  {
    "schema": "public",
    "function_name": "unlockrows",
    "arguments": "text",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.unlockrows(text)\n RETURNS
integer\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tret
int;\nBEGIN\n\n\tIF NOT LongTransactionsEnabled() THEN\n\t\tRAISE EXCEPTION 'Long
transaction support disabled, use EnableLongTransaction() to enable.';\n\tEND
IF;\n\n\tEXECUTE 'DELETE FROM authorization_table where authid = '
```

||\n\t\tquote_literal($1);\n\n\tGET DIAGNOSTICS ret = ROW_COUNT;\n\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "update_restaurant_product_preferences_updated_at",
    "arguments": "",
    "return_type": "trigger",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.update_restaurant_product_preferences_updated_at()\n RETURNS trigger\n LANGUAGE plpgsql\nAS $function$\nBEGIN\n NEW.updated_at = now();\n RETURN NEW;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "update_updated_at_column",
    "arguments": "",
    "return_type": "trigger",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.update_updated_at_column()\n RETURNS trigger\n LANGUAGE plpgsql\nAS $function$\nBEGIN\n   NEW.updated_at = NOW();\n   RETURN NEW;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "updategeometrysrid",
    "arguments": "catalogn_name character varying, schema_name character varying, table_name character varying, column_name character varying, new_srid_in integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.updategeometrysrid(catalogn_name character varying, schema_name character varying, table_name character varying, column_name character varying, new_srid_in integer)\n RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tmyrec RECORD;\n\tokay boolean;\n\tcname varchar;\n\treal_schema name;\n\tunknown_srid integer;\n\tnew_srid integer := new_srid_in;\n\nBEGIN\n\n\t-- Find, check or fix schema_name\n\tIF (schema_name != '') THEN\n\t\tokay = false;\n\n\t\tFOR myrec IN SELECT nspname FROM pg_namespace WHERE text(nspname) = schema_name LOOP\n\t\t\tokay := true;\n\t\tEND LOOP;\n\n\t\tIF (okay <> true) THEN\n\t\t\tRAISE EXCEPTION 'Invalid schema name';\n\t\tELSE\n\t\t\treal_schema = schema_name;\n\t\tEND IF;\n\tELSE\n\t\tSELECT INTO real_schema current_schema()::text;\n\tEND IF;\n\n\t-- Ensure that column_name is in geometry_columns\n\tokay = false;\n\tFOR myrec IN SELECT type, coord_dimension FROM public.geometry_columns WHERE f_table_schema = text(real_schema) and f_table_name = table_name and f_geometry_column = column_name LOOP\n\t\tokay := true;\n\tEND LOOP;\n\tIF (NOT okay) THEN\n\t\tRAISE EXCEPTION 'column not found in geometry_columns table';\n\t\tRETURN false;\n\tEND IF;\n\n\t-- Ensure that new_srid is

valid\n\tIF (new_srid > 0) THEN\n\t\tIF (SELECT count(*) = 0 from spatial_ref_sys where srid = new_srid) THEN\n\t\t\tRAISE EXCEPTION 'invalid SRID: % not found in spatial_ref_sys', new_srid;\n\t\t\tRETURN false;\n\t\tEND IF;\n\tELSE\n\t\tunknown_srid := public.ST_SRID('POINT EMPTY'::public.geometry);\n\t\tIF (new_srid != unknown_srid) THEN\n\t\t\tnew_srid := unknown_srid;\n\t\t\tRAISE NOTICE 'SRID value % converted to the officially unknown SRID value %', new_srid_in, new_srid;\n\t\tEND IF;\n\tEND IF;\n\n\tIF postgis_constraint_srid(real_schema, table_name, column_name) IS NOT NULL THEN\n\t-- srid was enforced with constraints before, keep it that way.\n\t\t-- Make up constraint name\n\t\tcname = 'enforce_srid_' || column_name;\n\n\t\t-- Drop enforce_srid constraint\n\t\tEXECUTE 'ALTER TABLE ' || quote_ident(real_schema) ||\n\t\t\t'.' || quote_ident(table_name) ||\n\t\t\t' DROP constraint ' || quote_ident(cname);\n\n\t\t-- Update geometries SRID\n\t\tEXECUTE 'UPDATE ' || quote_ident(real_schema) ||\n\t\t\t'.' || quote_ident(table_name) ||\n\t\t\t' SET ' || quote_ident(column_name) ||\n\t\t\t' = public.ST_SetSRID(' || quote_ident(column_name) ||\n\t\t\t', ' || new_srid::text || ')';\n\n\t\t-- Reset enforce_srid constraint\n\t\tEXECUTE 'ALTER TABLE ' || quote_ident(real_schema) ||\n\t\t\t'.' || quote_ident(table_name) ||\n\t\t\t' ADD constraint ' || quote_ident(cname) ||\n\t\t\t' CHECK (st_srid(' || quote_ident(column_name) ||\n\t\t\t') = ' || new_srid::text || ')';\n\tELSE\n\t\t-- We will use typmod to enforce if no srid constraints\n\t\t-- We are using postgis_type_name to lookup the new name\n\t\t-- (in case Paul changes his mind and flips geometry_columns to return old upper case name)\n\t\tEXECUTE 'ALTER TABLE ' || quote_ident(real_schema) || '.' || quote_ident(table_name) ||\n\t\t' ALTER COLUMN ' || quote_ident(column_name) || ' TYPE geometry(' || public.postgis_type_name(myrec.type, myrec.coord_dimension, true) || ', ' || new_srid::text || ') USING public.ST_SetSRID(' || quote_ident(column_name) || ',' || new_srid::text || ');' ;\n\tEND IF;\n\n\tRETURN real_schema || '.' || table_name || '.' || column_name ||' SRID changed to ' || new_srid::text;\n\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "updategeometrysrid",
    "arguments": "character varying, character varying, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.updategeometrysrid(character varying, character varying, integer)\n RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tret text;\nBEGIN\n\tSELECT public.UpdateGeometrySRID('','',$1,$2,$3) into ret;\n\tRETURN ret;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "updategeometrysrid",
    "arguments": "character varying, character varying, character varying, integer",
    "return_type": "text",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.updategeometrysrid(character varying, character varying, character varying, integer)\n RETURNS text\n LANGUAGE plpgsql\n STRICT\nAS $function$\nDECLARE\n\tret text;\nBEGIN\n\tSELECT public.UpdateGeometrySRID('',$1,$2,$3,$4) into ret;\n\tRETURN ret;\nEND;\n$function$\n"

```
  },
  {
    "schema": "public",
    "function_name": "vector",
    "arguments": "vector, integer, boolean",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector(vector, integer, boolean)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_accum",
    "arguments": "double precision[], vector",
    "return_type": "double precision[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_accum(double precision[], vector)\n RETURNS double precision[]\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_accum$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_add",
    "arguments": "vector, vector",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_add(vector, vector)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_add$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_avg",
    "arguments": "double precision[]",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_avg(double precision[])\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_avg$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_cmp",
    "arguments": "vector, vector",
    "return_type": "integer",
    "function_type": "function",
```

```
    "definition": "CREATE OR REPLACE FUNCTION public.vector_cmp(vector, vector)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_cmp$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_combine",
    "arguments": "double precision[], double precision[]",
    "return_type": "double precision[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_combine(double precision[], double precision[])\n RETURNS double precision[]\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_combine$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_concat",
    "arguments": "vector, vector",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_concat(vector, vector)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_concat$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_dims",
    "arguments": "vector",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_dims(vector)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_dims$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_dims",
    "arguments": "halfvec",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_dims(halfvec)\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$halfvec_vector_dims$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_eq",
```

```
    "arguments": "vector, vector",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_eq(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_eq$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_ge",
    "arguments": "vector, vector",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_ge(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_ge$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_gt",
    "arguments": "vector, vector",
    "return_type": "boolean",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_gt(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_gt$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_in",
    "arguments": "cstring, oid, integer",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_in(cstring, oid, integer)\n
RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_in$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_l2_squared_distance",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION
public.vector_l2_squared_distance(vector, vector)\n RETURNS double precision\n
LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector',
$function$vector_l2_squared_distance$function$\n"
  },
```

```
 {
   "schema": "public",
   "function_name": "vector_le",
   "arguments": "vector, vector",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.vector_le(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_le$function$\n"
 },
 {
   "schema": "public",
   "function_name": "vector_lt",
   "arguments": "vector, vector",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.vector_lt(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_lt$function$\n"
 },
 {
   "schema": "public",
   "function_name": "vector_mul",
   "arguments": "vector, vector",
   "return_type": "vector",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.vector_mul(vector, vector)\n
RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_mul$function$\n"
 },
 {
   "schema": "public",
   "function_name": "vector_ne",
   "arguments": "vector, vector",
   "return_type": "boolean",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION public.vector_ne(vector, vector)\n
RETURNS boolean\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS
'$libdir/vector', $function$vector_ne$function$\n"
 },
 {
   "schema": "public",
   "function_name": "vector_negative_inner_product",
   "arguments": "vector, vector",
   "return_type": "double precision",
   "function_type": "function",
   "definition": "CREATE OR REPLACE FUNCTION
public.vector_negative_inner_product(vector, vector)\n RETURNS double precision\n
```

LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_negative_inner_product$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_norm",
    "arguments": "vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_norm(vector)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_norm$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_out",
    "arguments": "vector",
    "return_type": "cstring",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_out(vector)\n RETURNS cstring\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_out$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_recv",
    "arguments": "internal, oid, integer",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_recv(internal, oid, integer)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_recv$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_search",
    "arguments": "query_embedding vector, search_term text DEFAULT ''::text, match_count integer DEFAULT 10",
    "return_type": "TABLE(id bigint, product_name text, similarity_distance double precision)",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_search(query_embedding vector, search_term text DEFAULT ''::text, match_count integer DEFAULT 10)\n RETURNS TABLE(id bigint, product_name text, similarity_distance double precision)\n LANGUAGE plpgsql\n SECURITY DEFINER\nAS $function$\nBEGIN\n RETURN QUERY\n SELECT \n ml.id,\n   ml.product_name::text,\n   (ml.embedding_vector_v2 <=> query_embedding)::float as similarity_distance\n FROM master_list ml\n WHERE ml.is_active = true \n   AND ml.embedding_vector_v2 IS NOT NULL\n   AND (\n   search_term = '' OR\n

ml.product_name ILIKE '%' || search_term || '%'\n   )\n ORDER BY similarity_distance ASC\n LIMIT match_count;\nEND;\n$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_send",
    "arguments": "vector",
    "return_type": "bytea",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_send(vector)\n RETURNS bytea\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_send$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_spherical_distance",
    "arguments": "vector, vector",
    "return_type": "double precision",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_spherical_distance(vector, vector)\n RETURNS double precision\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_spherical_distance$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_sub",
    "arguments": "vector, vector",
    "return_type": "vector",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_sub(vector, vector)\n RETURNS vector\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_sub$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_to_float4",
    "arguments": "vector, integer, boolean",
    "return_type": "real[]",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_to_float4(vector, integer, boolean)\n RETURNS real[]\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_to_float4$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_to_halfvec",
    "arguments": "vector, integer, boolean",
    "return_type": "halfvec",

```
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_to_halfvec(vector, integer, boolean)\n RETURNS halfvec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_to_halfvec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_to_sparsevec",
    "arguments": "vector, integer, boolean",
    "return_type": "sparsevec",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_to_sparsevec(vector, integer, boolean)\n RETURNS sparsevec\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_to_sparsevec$function$\n"
  },
  {
    "schema": "public",
    "function_name": "vector_typmod_in",
    "arguments": "cstring[]",
    "return_type": "integer",
    "function_type": "function",
    "definition": "CREATE OR REPLACE FUNCTION public.vector_typmod_in(cstring[])\n RETURNS integer\n LANGUAGE c\n IMMUTABLE PARALLEL SAFE STRICT\nAS '$libdir/vector', $function$vector_typmod_in$function$\n"
  }
]
```


8. Policies RLS (Row Level Security)


```
[
  {
    "schemaname": "public",
    "tablename": "line_sessions",
    "policyname": "Allow authenticated read access",
    "permissive": "PERMISSIVE",
    "roles": "{authenticated}",
    "command": "SELECT",
    "using_expression": "true",
    "with_check_expression": null
  },
  {
    "schemaname": "public",
    "tablename": "master_list",
    "policyname": "Allow authenticated read access",
    "permissive": "PERMISSIVE",
    "roles": "{authenticated}",
    "command": "SELECT",
```

```
  "using_expression": "true",
  "with_check_expression": null
},
{
  "schemaname": "public",
  "tablename": "pricing_history",
  "policyname": "Allow authenticated read access",
  "permissive": "PERMISSIVE",
  "roles": "{authenticated}",
  "command": "SELECT",
  "using_expression": "true",
  "with_check_expression": null
},
{
  "schemaname": "public",
  "tablename": "product_categories",
  "policyname": "Allow authenticated read access",
  "permissive": "PERMISSIVE",
  "roles": "{authenticated}",
  "command": "SELECT",
  "using_expression": "true",
  "with_check_expression": null
},
{
  "schemaname": "public",
  "tablename": "purchase_orders",
  "policyname": "Allow authenticated read access",
  "permissive": "PERMISSIVE",
  "roles": "{authenticated}",
  "command": "SELECT",
  "using_expression": "true",
  "with_check_expression": null
},
{
  "schemaname": "public",
  "tablename": "restaurant_people",
  "policyname": "Allow authenticated read access",
  "permissive": "PERMISSIVE",
  "roles": "{authenticated}",
  "command": "SELECT",
  "using_expression": "true",
  "with_check_expression": null
},
{
  "schemaname": "public",
  "tablename": "restaurants",
  "policyname": "Allow authenticated read access",
  "permissive": "PERMISSIVE",
```

```
    "roles": "{authenticated}",
    "command": "SELECT",
    "using_expression": "true",
    "with_check_expression": null
  },
  {
    "schemaname": "public",
    "tablename": "supplier_mapped_products",
    "policyname": "Allow authenticated read access",
    "permissive": "PERMISSIVE",
    "roles": "{authenticated}",
    "command": "SELECT",
    "using_expression": "true",
    "with_check_expression": null
  },
  {
    "schemaname": "public",
    "tablename": "suppliers",
    "policyname": "Allow authenticated read access",
    "permissive": "PERMISSIVE",
    "roles": "{authenticated}",
    "command": "SELECT",
    "using_expression": "true",
    "with_check_expression": null
  }
]
```


9. Extensiones Habilitadas


```
[
  {
    "extension_name": "pg_graphql",
    "version": "1.5.11",
    "relocatable": false
  },
  {
    "extension_name": "pg_stat_statements",
    "version": "1.11",
    "relocatable": true
  },
  {
    "extension_name": "pgcrypto",
    "version": "1.3",
    "relocatable": true
  },
  {
    "extension_name": "plpgsql",
```

```
    "version": "1.0",
    "relocatable": false
  },
  {
    "extension_name": "postgis",
    "version": "3.3.7",
    "relocatable": false
  },
  {
    "extension_name": "supabase_vault",
    "version": "0.3.1",
    "relocatable": false
  },
  {
    "extension_name": "uuid-ossp",
    "version": "1.1",
    "relocatable": true
  },
  {
    "extension_name": "vector",
    "version": "0.8.0",
    "relocatable": true
  }
]
```


10. Constraints (CHECK, UNIQUE, etc.)


```
[
  {
    "table_name": "line_sessions",
    "constraint_name": "2200_19034_1_not_null",
    "constraint_type": "CHECK",
    "check_clause": "session_id IS NOT NULL",
    "column_name": null
  },
  {
    "table_name": "master_list",
    "constraint_name": "2200_18915_2_not_null",
    "constraint_type": "CHECK",
    "check_clause": "product_name IS NOT NULL",
    "column_name": null
  },
  {
    "table_name": "master_list",
    "constraint_name": "2200_18915_1_not_null",
    "constraint_type": "CHECK",
    "check_clause": "id IS NOT NULL",
```

```
  "column_name": null
},
{
  "table_name": "pricing_history",
  "constraint_name": "2200_19106_10_not_null",
  "constraint_type": "CHECK",
  "check_clause": "effective_date IS NOT NULL",
  "column_name": null
},
{
  "table_name": "pricing_history",
  "constraint_name": "2200_19106_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "pricing_history",
  "constraint_name": "2200_19106_5_not_null",
  "constraint_type": "CHECK",
  "check_clause": "unit_price IS NOT NULL",
  "column_name": null
},
{
  "table_name": "product_categories",
  "constraint_name": "2200_18882_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "product_categories",
  "constraint_name": "2200_18882_2_not_null",
  "constraint_type": "CHECK",
  "check_clause": "category_name IS NOT NULL",
  "column_name": null
},
{
  "table_name": "product_categories",
  "constraint_name": "product_categories_category_slug_key",
  "constraint_type": "UNIQUE",
  "check_clause": null,
  "column_name": "category_slug"
},
{
  "table_name": "purchase_orders",
  "constraint_name": "2200_19073_1_not_null",
  "constraint_type": "CHECK",
```

```
  "check_clause": "order_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_people",
  "constraint_name": "2200_19003_5_not_null",
  "constraint_type": "CHECK",
  "check_clause": "last_name IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_people",
  "constraint_name": "2200_19003_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_people",
  "constraint_name": "2200_19003_4_not_null",
  "constraint_type": "CHECK",
  "check_clause": "first_name IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences",
  "constraint_name": "2200_40631_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences",
  "constraint_name": "2200_40631_3_not_null",
  "constraint_type": "CHECK",
  "check_clause": "master_list_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences",
  "constraint_name": "2200_40631_2_not_null",
  "constraint_type": "CHECK",
  "check_clause": "restaurant_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences",
  "constraint_name": "restaurant_product_preferences_unique",
```

```
  "constraint_type": "UNIQUE",
  "check_clause": null,
  "column_name": "restaurant_id"
},
{
  "table_name": "restaurant_product_preferences",
  "constraint_name": "restaurant_product_preferences_unique",
  "constraint_type": "UNIQUE",
  "check_clause": null,
  "column_name": "master_list_id"
},
{
  "table_name": "restaurant_product_preferences_history",
  "constraint_name": "2200_40699_3_not_null",
  "constraint_type": "CHECK",
  "check_clause": "restaurant_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences_history",
  "constraint_name": "2200_40699_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences_history",
  "constraint_name": "2200_40699_2_not_null",
  "constraint_type": "CHECK",
  "check_clause": "preference_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences_history",
  "constraint_name": "2200_40699_5_not_null",
  "constraint_type": "CHECK",
  "check_clause": "change_type IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurant_product_preferences_history",
  "constraint_name": "2200_40699_4_not_null",
  "constraint_type": "CHECK",
  "check_clause": "master_list_id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurants",
```

```
  "constraint_name": "2200_18987_7_not_null",
  "constraint_type": "CHECK",
  "check_clause": "street_address IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurants",
  "constraint_name": "2200_18987_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "restaurants",
  "constraint_name": "2200_18987_2_not_null",
  "constraint_type": "CHECK",
  "check_clause": "restaurant_name IS NOT NULL",
  "column_name": null
},
{
  "table_name": "spatial_ref_sys",
  "constraint_name": "spatial_ref_sys_srid_check",
  "constraint_type": "CHECK",
  "check_clause": null,
  "column_name": null
},
{
  "table_name": "spatial_ref_sys",
  "constraint_name": "2200_17605_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": null,
  "column_name": null
},
{
  "table_name": "supplier_mapped_products",
  "constraint_name": "2200_18964_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "supplier_mapped_products",
  "constraint_name": "2200_18964_45_not_null",
  "constraint_type": "CHECK",
  "check_clause": "supplier_id IS NOT NULL",
  "column_name": null
},
{
```

```
  "table_name": "supplier_mapped_products",
  "constraint_name": "2200_18964_14_not_null",
  "constraint_type": "CHECK",
  "check_clause": "current_unit_price IS NOT NULL",
  "column_name": null
},
{
  "table_name": "supplier_mapped_products",
  "constraint_name": "2200_18964_8_not_null",
  "constraint_type": "CHECK",
  "check_clause": "mapping_confidence IS NOT NULL",
  "column_name": null
},
{
  "table_name": "supplier_mapped_products",
  "constraint_name": "2200_18964_3_not_null",
  "constraint_type": "CHECK",
  "check_clause": "supplier_product_code IS NOT NULL",
  "column_name": null
},
{
  "table_name": "suppliers",
  "constraint_name": "2200_18936_2_not_null",
  "constraint_type": "CHECK",
  "check_clause": "company_name IS NOT NULL",
  "column_name": null
},
{
  "table_name": "suppliers",
  "constraint_name": "2200_18936_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
{
  "table_name": "user_preferences",
  "constraint_name": "2200_40425_4_not_null",
  "constraint_type": "CHECK",
  "check_clause": "preference_type IS NOT NULL",
  "column_name": null
},
{
  "table_name": "user_preferences",
  "constraint_name": "2200_40425_1_not_null",
  "constraint_type": "CHECK",
  "check_clause": "id IS NOT NULL",
  "column_name": null
},
```

```
  {
    "table_name": "user_preferences",
    "constraint_name": "2200_40425_7_not_null",
    "constraint_type": "CHECK",
    "check_clause": "preference_value IS NOT NULL",
    "column_name": null
  }
]
```