Задание 7 "Основы SQL и СУБД (1)"

Поток 1. `dim_date` — генерация календаря

```
INSERT INTO dim_date (
    date_id,
    full_date,
    day,
    month,
    year,
    quarter,
    month_name,
    is_weekend
)
SELECT
    TO_CHAR(datum, 'YYYYMMDD')::INT AS date_id,
    datum AS full_date,
    EXTRACT(DAY FROM datum)::INT AS day,
    EXTRACT(MONTH FROM datum)::INT AS month,
    EXTRACT(YEAR FROM datum)::INT AS year,
    CEIL(EXTRACT(MONTH FROM datum) / 3.0)::INT AS quarter,
    TO_CHAR(datum, 'Month') AS month_name,
    CASE WHEN EXTRACT(DOW FROM datum) IN (0,6) THEN TRUE ELSE FALSE END AS
is_weekend
FROM (
    SELECT '2020-01-01'::DATE + SEQUENCE.DAY AS datum
    FROM GENERATE_SERIES(0, 365*11) AS SEQUENCE(DAY)
) DQ
WHERE NOT EXISTS (
    SELECT 1 FROM dim_date WHERE date_id = TO_CHAR(datum, 'YYYYMMDD')::INT
);
```

Поток 2. `dim_channel` — справочник каналов (Full Extract)

```
INSERT INTO dim_channel (
    channel_id,
    channel_cd,
    channel_name
)
SELECT
    c.channel_id,
    c.channel_cd,
    c.channel_name
FROM channel c
WHERE NOT EXISTS (
    SELECT 1 FROM dim_channel dc WHERE dc.channel_id = c.channel_id
);
```

Поток 3. `dim_product_type` — типы продуктов (DISTINCT)

```
INSERT INTO dim_product_type (
    product_type_name
)
SELECT DISTINCT
```

```
    p.product_type AS product_type_name
FROM product p
WHERE p.product_type IS NOT NULL
  AND NOT EXISTS (
      SELECT 1 FROM dim_product_type pt WHERE pt.product_type_name = p.product_type
  );
```

Поток 4. `dim_product` — справочник продуктов (Incremental)

```
INSERT INTO dim_product (
    product_id,
    product_type_id,
    product_name,
    description,
    brand,
    category,
    launch_date,
    status
)
SELECT
    p.product_id,
    pt.product_type_id,
    p.product_name,
    p.description,
    p.brand,
    p.category,
    p.launch_date::DATE,
    p.status
FROM product p
JOIN dim_product_type pt ON pt.product_type_name = p.product_type
WHERE NOT EXISTS (
    SELECT 1 FROM dim_product dp WHERE dp.product_id = p.product_id
)
AND p.updated_at > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name = 'dim_product'),
'1970-01-01'
);
```

Поток 5. `dim_product_terms` — SCD Type 2 (история условий)

```
-- закрыть предыдущие версии, если появились новые
UPDATE dim_product_terms
SET valid_to_dttm = NOW() - INTERVAL '1 second',
    is_current = FALSE
WHERE (product_id, valid_from_dttm) IN (
    SELECT pa.product_id, MIN(pa.valid_from_dttm)
    FROM product_audit pa
    JOIN dim_product_terms dpt
      ON dpt.product_id = pa.product_id
     AND dpt.is_current = TRUE
     AND dpt.valid_from_dttm < pa.valid_from_dttm
    GROUP BY pa.product_id
);

-- вставить новые версии
```

```
INSERT INTO dim_product_terms (
    product_id,
    interest_rate,
    min_amount,
    max_amount,
    valid_from_dttm,
    valid_to_dttm,
    is_current
)
SELECT
    pa.product_id,
    pa.interest_rate,
    pa.min_amount,
    pa.max_amount,
    pa.valid_from_dttm,
    pa.valid_to_dttm,
    pa.is_current
FROM product_audit pa
WHERE NOT EXISTS (
    SELECT 1
    FROM dim_product_terms dpt
    WHERE dpt.product_id = pa.product_id
      AND dpt.valid_from_dttm = pa.valid_from_dttm
)
AND pa.created_at > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name =
'dim_product_terms'),
    '1970-01-01'
);
```

Поток 6. `dim_customer` — клиенты (Incremental)

```
INSERT INTO dim_customer (
    customer_id,
    first_name,
    last_name,
    birth_date,
    passport_series,
    passport_number
)
SELECT
    c.client_id,
    c.first_name,
    c.last_name,
    c.birth_date::DATE,
    c.passport_series,
    c.passport_number
FROM client c
WHERE NOT EXISTS (
    SELECT 1 FROM dim_customer dc WHERE dc.customer_id = c.client_id
)
AND c.updated_at > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name = 'dim_customer'),
    '1970-01-01'
);
```

Поток 7. `dim_account` — счета (Incremental)

```
INSERT INTO dim_account (
    account_id,
    account_number,
    currency,
    open_date,
    close_date,
    product_id,
    customer_id
)
SELECT
    a.account_id,
    a.account_number,
    a.currency,
    a.open_date::DATE,
    a.close_date::DATE,
    a.product_id,
    a.client_id
FROM account a
WHERE NOT EXISTS (
    SELECT 1 FROM dim_account da WHERE da.account_id = a.account_id
)
AND a.updated_at > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name = 'dim_account'),
    '1970-01-01'
);
```

Поток 8. `fact_product_activation` — факт подключения (Incremental)

```
INSERT INTO fact_product_activation (
    activation_id,
    customer_id,
    product_id,
    channel_id,
    activation_date_id,
    product_terms_id,
    quantity,
    status
)
SELECT
    pa.application_id,
    pa.client_id,
    pa.product_id,
    pa.channel_id,
    TO_CHAR(pa.activation_dttm, 'YYYYMMDD')::INT,
    dpt.product_terms_id,
    1 AS quantity,
    pa.status
FROM product_application pa
JOIN dim_product_terms dpt
  ON dpt.product_id = pa.product_id
 AND pa.activation_dttm >= dpt.valid_from_dttm
 AND (pa.activation_dttm < dpt.valid_to_dttm OR dpt.valid_to_dttm IS NULL)
WHERE NOT EXISTS (
```

```
    SELECT 1 FROM fact_product_activation f WHERE f.activation_id =
pa.application_id
)
AND pa.activation_dttm::DATE > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name =
'fact_product_activation'),
    '1970-01-01'
);
```

Поток 9. `fact_transaction` — финансовые операции (Incremental)

```
INSERT INTO fact_transaction (
    transaction_id,
    account_id,
    amount,
    transaction_date_id,
    transaction_type
)
SELECT
    t.transaction_id,
    t.account_id,
    t.amount,
    TO_CHAR(t.transaction_dttm, 'YYYYMMDD')::INT,
    t.transaction_type
FROM transaction t
WHERE NOT EXISTS (
    SELECT 1 FROM fact_transaction ft WHERE ft.transaction_id = t.transaction_id
)
AND t.transaction_dttm::DATE > COALESCE(
    (SELECT MAX(load_timestamp) FROM etl_log WHERE table_name =
'fact_transaction'),
    '1970-01-01'
);
```