```
{
 "GIG4U_User_Schemas": {

  "1_CORE_IDENTITY": {
   "table": "users",
   "description": "Single table for ALL actor types. user_type decides who they are.",
   "fields": {
    "id":              "UUID (PK) — UUIDv7",
    "phone":              "VARCHAR(15) UNIQUE NOT NULL",
    "email":             "VARCHAR(255) UNIQUE",
    "password_hash":      "TEXT",
    "user_type":          "'CLIENT' | 'SP' | 'ADMIN' | 'PARTNER'",
    "status":            "'ACTIVE' | 'INACTIVE' | 'BANNED' | 'SUSPENDED'",
    "is_phone_verified":  "BOOLEAN DEFAULT false",
    "is_email_verified":  "BOOLEAN DEFAULT false",
    "last_login_at":      "TIMESTAMP",
    "created_at":         "TIMESTAMP",
    "updated_at":         "TIMESTAMP",
    "deleted_at":         "TIMESTAMP NULL  ← soft delete"
   }
  },

  "2_PROFILES": {

   "client_profiles": {
    "table": "client_profiles",
    "linked_to": "users.id (1:1)",
    "fields": {
     "user_id":       "UUID (PK + FK → users.id)",
     "tenant_id":      "UUID (FK → tenants.id)  ← which company they belong to",
     "full_name":      "VARCHAR(255)",
     "designation":   "VARCHAR(100)  e.g. 'HR Manager'",
     "department":    "VARCHAR(100)  e.g. 'Operations'",
     "client_role":   "'ADMIN' | 'MANAGER' | 'FINANCE' | 'VIEWER'  ← within their company",
     "created_at":    "TIMESTAMP",
     "updated_at":    "TIMESTAMP"
    }
   },

   "service_provider_profiles": {
    "table": "service_provider_profiles",
    "linked_to": "users.id (1:1)",
    "fields": {
     "user_id":          "UUID (PK + FK → users.id)",
     "full_name":         "VARCHAR(255)",
     "city":              "VARCHAR(100)",
     "state":             "VARCHAR(100)",
     "pincode":           "VARCHAR(10)",
     "gender":            "VARCHAR(20)",
     "date_of_birth":     "DATE",
     "profile_photo_url": "TEXT",
     "sp_status":         "'PROFILE_INCOMPLETE' | 'KYC_PENDING' | 'KYC_SUBMITTED' |
'KYC_APPROVED' | 'ACTIVE' | 'SUSPENDED' | 'BANNED'",
     "behavior_score":    "NUMERIC(5,2) DEFAULT 0",
     "rating_avg":        "NUMERIC(3,2) DEFAULT 0",
     "total_completed":   "INT DEFAULT 0",
     "created_at":        "TIMESTAMP",
     "updated_at":        "TIMESTAMP"
    }
   },

   "admin_profiles": {
    "table": "admin_profiles",
    "linked_to": "users.id (1:1)",
    "fields": {
     "user_id":        "UUID (PK + FK → users.id)",
     "full_name":      "VARCHAR(255)",
     "employee_id":    "VARCHAR(100) UNIQUE",
     "department":     "VARCHAR(100)  e.g. 'KYC Team', 'Finance'",
     "active_role_id": "UUID (FK → roles.id)  ← last selected role from dialog",
     "created_at":     "TIMESTAMP",
     "updated_at":     "TIMESTAMP"
    }
   },

   "partner_profiles__FUTURE": {
    "table": "partner_profiles",
    "linked_to": "users.id (1:1)",
    "note": "Activate by adding 'PARTNER' to users.user_type enum + run migration",
    "fields": {
     "user_id":        "UUID (PK + FK → users.id)",
     "organization":   "VARCHAR(255)",
     "partner_type":   "'AGGREGATOR' | 'RECRUITER' | 'FRANCHISE'",
     "scope_cities":   "TEXT[]  ← array of cities they can operate in",
     "commission_pct": "NUMERIC(5,2)",
     "created_at":     "TIMESTAMP",
     "updated_at":     "TIMESTAMP"
    }
   },

   "new_place_user_profiles__EXAMPLE_FUTURE": {
    "table": "new_place_user_profiles",
    "linked_to": "users.id (1:1)",
    "note": "Template — copy this pattern for ANY new actor type",
    "fields": {
     "user_id":      "UUID (PK + FK → users.id)",
     "full_name":    "VARCHAR(255)",
     "custom_field": "add whatever fields this actor needs",
     "created_at":   "TIMESTAMP",
     "updated_at":   "TIMESTAMP"
    }
   }

 },

 "3_RBAC_CORE": {

   "roles": {
    "table": "roles",
    "description": "All roles live here — system seeded + dynamically created by Super
Admin",
    "fields": {
     "id":           "UUID (PK)",
     "name":         "VARCHAR(100) UNIQUE  e.g. 'KYC_ADMIN', 'CLIENT_MANAGER'",
     "display_name": "VARCHAR(255)  e.g. 'KYC & Verification Admin'",
     "description":  "TEXT",
     "actor_type":   "'ADMIN' | 'CLIENT' | 'SP' | 'PARTNER'  ← which user_type this role
belongs to",
     "parent_id":    "UUID (FK → roles.id)  ← NULL for top-level, set for sub-roles",
     "is_system":    "BOOLEAN  ← true = cannot be deleted (seeded roles)",
     "is_active":    "BOOLEAN DEFAULT true",
     "created_by":   "UUID (FK → users.id)  ← which Super Admin created this",
     "created_at":   "TIMESTAMP",
     "updated_at":   "TIMESTAMP",
     "deleted_at":   "TIMESTAMP NULL  ← soft delete"
    },
    "seeded_roles": [
     "SUPER_ADMIN  (actor_type: ADMIN, parent_id: null)",
     "KYC_ADMIN  (actor_type: ADMIN, parent_id: SUPER_ADMIN)",
     "MESSAGE_ADMIN  (actor_type: ADMIN, parent_id: SUPER_ADMIN)",
     "FINANCE_ADMIN  (actor_type: ADMIN, parent_id: SUPER_ADMIN)",
     "OPERATIONS_ADMIN  (actor_type: ADMIN, parent_id: SUPER_ADMIN)",
     "SUPPORT_ADMIN  (actor_type: ADMIN, parent_id: SUPER_ADMIN)",
     "CLIENT_ADMIN  (actor_type: CLIENT)",
     "CLIENT_MANAGER  (actor_type: CLIENT)",
     "CLIENT_VIEWER  (actor_type: CLIENT)",
     "SP  (actor_type: SP)"
    ]
   },

   "permission_groups": {
    "table": "permission_groups",
    "description": "Groups permissions by feature module — for clean UI display in admin
panel",
    "fields": {
     "id":           "UUID (PK)",
     "name":         "VARCHAR(100) UNIQUE  e.g. 'kyc', 'billing', 'roles'",
     "display_name": "VARCHAR(255)  e.g. 'KYC & Identity Verification'",
     "description":  "TEXT",
     "created_at":   "TIMESTAMP"
    },
    "seeded_groups": [
     "kyc", "users", "roles", "projects",
     "billing", "messaging", "analytics", "sp_management"
    ]
   },

   "permissions": {
    "table": "permissions",
    "description": "Granular permissions using resource:action pattern",
    "fields": {
     "id":           "UUID (PK)",
     "group_id":     "UUID (FK → permission_groups.id)",
     "name":         "VARCHAR(150) UNIQUE  e.g. 'kyc:approve', 'users:ban', 'roles:create'",
     "display_name": "VARCHAR(255)  e.g. 'Approve KYC'",
     "description":  "TEXT",
     "is_active":    "BOOLEAN DEFAULT true",
     "created_at":   "TIMESTAMP"
    },
    "seeded_permissions_examples": {
     "kyc":          ["kyc:view", "kyc:approve", "kyc:reject", "kyc:flag"],
     "users":        ["users:list", "users:view", "users:ban", "users:delete"],
     "roles":        ["roles:create", "roles:edit", "roles:delete", "roles:assign"],
     "projects":     ["projects:list", "projects:create", "projects:approve", "projects:close"],
     "billing":      ["billing:view", "billing:process_payout", "billing:generate_invoice"],
     "messaging":    ["messaging:send_broadcast", "messaging:view_logs"],
      "analytics":    ["analytics:view_dashboard", "analytics:export"],
      "sp_management":["sp:onboard", "sp:suspend", "sp:view_score"]
    }
  },

  "role_permissions": {
    "table": "role_permissions",
    "description": "Which permissions are assigned to which role",
    "fields": {
      "role_id":       "UUID (FK → roles.id)  ← composite PK part 1",
      "permission_id": "UUID (FK → permissions.id)  ← composite PK part 2",
      "granted_by":    "UUID (FK → users.id)  ← Super Admin who assigned it",
      "granted_at":    "TIMESTAMP"
    },
    "example": {
      "KYC_ADMIN role gets":      ["kyc:view", "kyc:approve", "kyc:reject"],
      "FINANCE_ADMIN role gets":  ["billing:view", "billing:process_payout",
"billing:generate_invoice"],
      "SUPER_ADMIN role gets":    "ALL permissions (wildcard)"
    }
  },

  "user_roles": {
    "table": "user_roles",
    "description": "Which roles are assigned to which user — scoped by tenant for CLIENT
roles",
    "fields": {
      "id":           "UUID (PK)",
      "user_id":      "UUID (FK → users.id)",
      "role_id":      "UUID (FK → roles.id)",
      "tenant_id":    "UUID (FK → tenants.id) NULL  ← NULL for ADMIN/SP, set for CLIENT
roles",
      "assigned_by":  "UUID (FK → users.id)  ← who assigned",
      "assigned_at":  "TIMESTAMP",
      "expires_at":   "TIMESTAMP NULL  ← optional time-bound role",
      "is_active":    "BOOLEAN DEFAULT true"
    },
    "scoping_examples": {
      "Admin user":  "tenant_id = NULL  (platform-wide)",
      "SP user":     "tenant_id = NULL  (platform-wide)",
      "Client user": "tenant_id = 'abc-123'  (scoped to their company only)"
    }
  }
  },

  "4_HOW_TO_ADD_NEW_ROLE": {
   "example": "Adding a new admin sub-role called CONTENT_ADMIN",
   "steps": {
    "step_1_insert_role": {
     "table": "roles",
     "action": "INSERT",
     "data": {
      "name":        "CONTENT_ADMIN",
      "display_name":"Content & Media Admin",
      "actor_type":  "ADMIN",
      "parent_id":   "<SUPER_ADMIN role UUID>",
      "is_system":   false
     }
    },
    "step_2_assign_permissions": {
     "table": "role_permissions",
     "action": "INSERT rows",
     "data": [
      { "role_id": "<CONTENT_ADMIN UUID>", "permission_id": "<analytics:view_dashboard
UUID>" },
      { "role_id": "<CONTENT_ADMIN UUID>", "permission_id":
"<messaging:send_broadcast UUID>" }
     ]
    },
    "step_3_assign_to_admin_user": {
     "table": "user_roles",
     "action": "INSERT",
     "data": {
      "user_id":  "<admin user UUID>",
      "role_id":  "<CONTENT_ADMIN UUID>",
      "tenant_id": null
     }
    },
    "no_code_changes_needed": true,
    "no_migration_needed": true
   }
  },

  "5_HOW_TO_ADD_NEW_PERMISSION": {
   "example": "Adding a new permission called content:publish",
   "steps": {
    "step_1_optionally_create_group": {
        "table": "permission_groups",
        "action": "INSERT (only if new module)",
        "data": {
          "name":        "content",
          "display_name": "Content Management"
        }
      },
      "step_2_insert_permission": {
        "table": "permissions",
        "action": "INSERT",
        "data": {
          "group_id":     "<content group UUID>",
          "name":        "content:publish",
          "display_name": "Publish Content",
          "description":  "Allows publishing articles and media"
        }
      },
      "step_3_assign_to_roles": {
        "table": "role_permissions",
        "action": "INSERT rows for each role that should have it",
        "data": [
          { "role_id": "<CONTENT_ADMIN UUID>", "permission_id": "<content:publish UUID>" },
          { "role_id": "<SUPER_ADMIN UUID>",   "permission_id": "<content:publish UUID>" }
        ]
      },
      "no_code_changes_needed": true
    }
  },

  "6_HOW_TO_ADD_NEW_ACTOR_TYPE": {
    "example": "Adding newPlaceUser as a 4th login type",
    "steps": {
      "step_1_alter_users_table": "ALTER TABLE users DROP CONSTRAINT
users_user_type_check; ALTER TABLE users ADD CONSTRAINT users_user_type_check
CHECK (user_type IN ('CLIENT','SP','ADMIN','PARTNER','NEW_PLACE_USER'));",
      "step_2_create_profile_table": "CREATE TABLE new_place_user_profiles (user_id UUID
PK FK users.id, ...your fields...)",
      "step_3_add_nestjs_module": "Create NewPlaceUserProfileModule in NestJS (no changes
to Auth/Role/Permission modules)",
      "step_4_seed_role": {
        "table": "roles",
        "data": { "name": "NEW_PLACE_USER", "actor_type": "NEW_PLACE_USER",
"is_system": true }
      },
      "step_5_seed_permissions": "INSERT relevant permissions into permissions table",
      "step_6_done": "Login screen now supports 4 options — zero changes to RBAC core
tables"
    }
  }

}
}
```