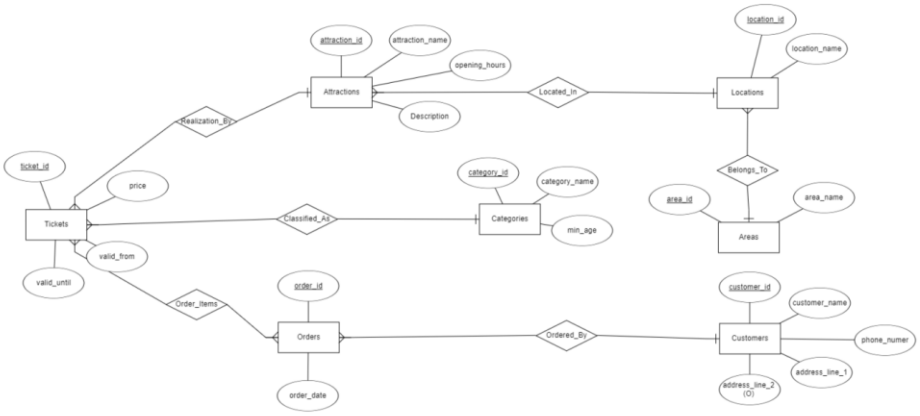
Tickets Online Store

Presenter: Bina Cohen | 207562901

ERD:



Entities:

1 .Categories: This table stores different categories that can be assigned to tickets. Each category has an ID, a name (like "Adult" or "Child"), and a minimum age requirement.

2 .Customers: This table stores information about customers who purchase tickets. Each customer has an ID, a name, a phone number, and an address (consisting of two lines).

3 .Orders: This table keeps track of orders placed by customers. Each order has a total price, a date, and a unique ID. It also references the customer who placed the order.

4 .Areas: This table represents different areas within a location. Each area has an ID and a name.

5 .Locations: This table stores information about locations where attractions are located. Each location has an ID, a name, and is associated with an area.

6 .Attractions: This table contains details about attractions available at various locations. Each attraction has an ID, a name, a description, opening hours, and is associated with a location.

7 .Tickets: This table stores information about tickets that customers purchase for attractions. Each ticket has an ID, a price, validity dates, and is associated with a category and an attraction.

8 .Order_Items: This table represents individual items within an order. Each order item references a ticket and an order, forming a link between the tickets purchased and the orders they belong to.

These tables together form a relational database schema designed to manage customer orders for attractions, including ticket purchases and associated information such as categories, locations, and attractions.

Relationships:

1. Customers and Orders:

One customer can place multiple orders. This is a one-to-many relationship because a single customer can place many orders, but each order is associated with only one customer.

The `customer_id` in the Orders table serves as a foreign key referencing the Customers table, establishing this relationship.

2. Orders and Order_Items:

Each order can contain multiple order items. This is a many-to-many relationship because one order can have multiple tickets (order items), and one ticket can belong to multiple orders.

The combination of `ticket_id` and `order_id` in the Order_Items table serves as a composite primary key, referencing both the Tickets and Orders tables.

3. Tickets and Categories:

Each ticket belongs to a specific category. This is a many-to-one relationship because multiple tickets can belong to the same category, but each ticket is associated with only one category.

The `category_id` in the Tickets table serves as a foreign key referencing the Categories table.

4. Tickets and Attractions:

Each ticket is valid for a specific attraction. This is a many-to-one relationship because multiple tickets can be valid for the same attraction, but each ticket is associated with only one attraction.

The `attraction_id` in the Tickets table serves as a foreign key referencing the Attractions table.

5. Attractions and Locations:

Each attraction is located at a specific location. This is a many-to-one relationship because multiple attractions can be located at the same location, but each attraction is associated with only one location.

The `location_id` in the Attractions table serves as a foreign key referencing the Locations table.

6. Locations and Areas:

Each location belongs to a specific area. This is a many-to-one relationship because multiple locations can belong to the same area, but each location is associated with only one area.

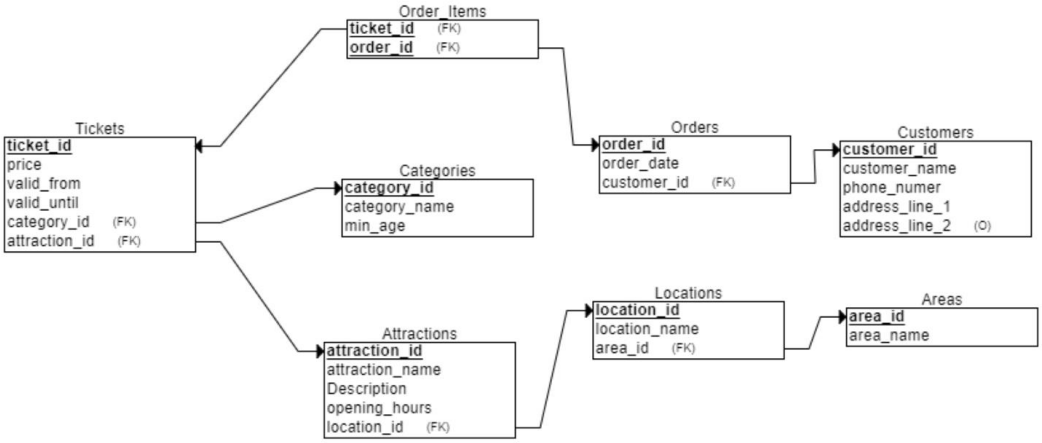
The `area_id` in the Locations table serves as a foreign key referencing the Areas table.

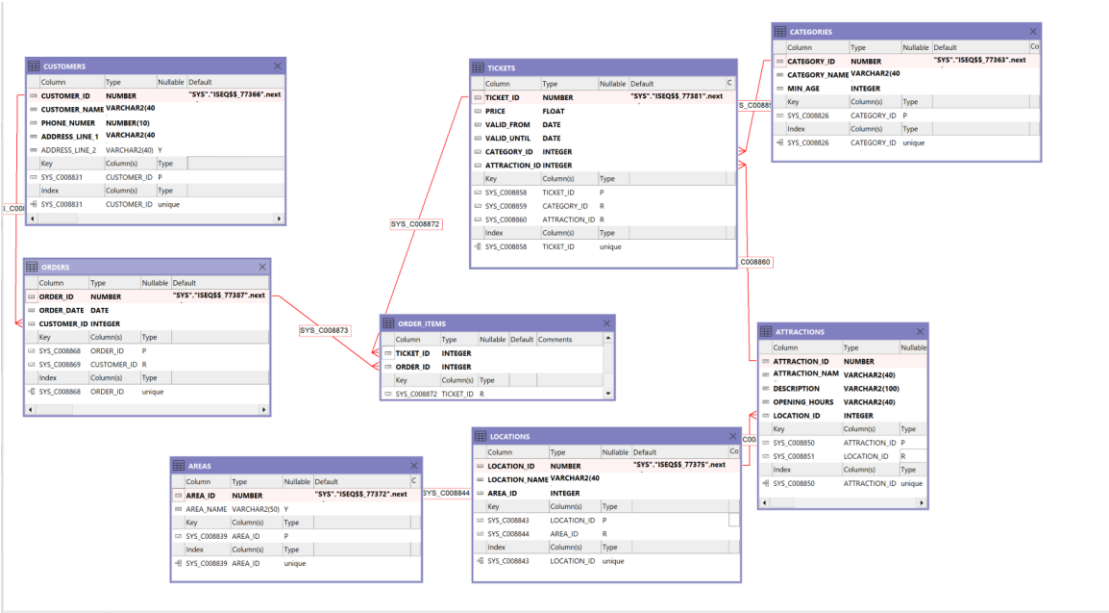
These relationships help organize and establish connections between the entities in the database, enabling efficient data retrieval and management.

Normal Form (3NF):

The database schema has been validated and confirmed to be in Third Normal Form (3NF). Each table is structured with a well-defined primary key, and all non-key attributes are directly associated with the primary key, without any transitive dependencies.

DSD Diagram:





SQL Code:

```
CREATE TABLE Categories
(
 category_id INT NOT NULL,
 category_name INT NOT NULL,
 min_age INT NOT NULL,
 PRIMARY KEY (category_id)
);
```

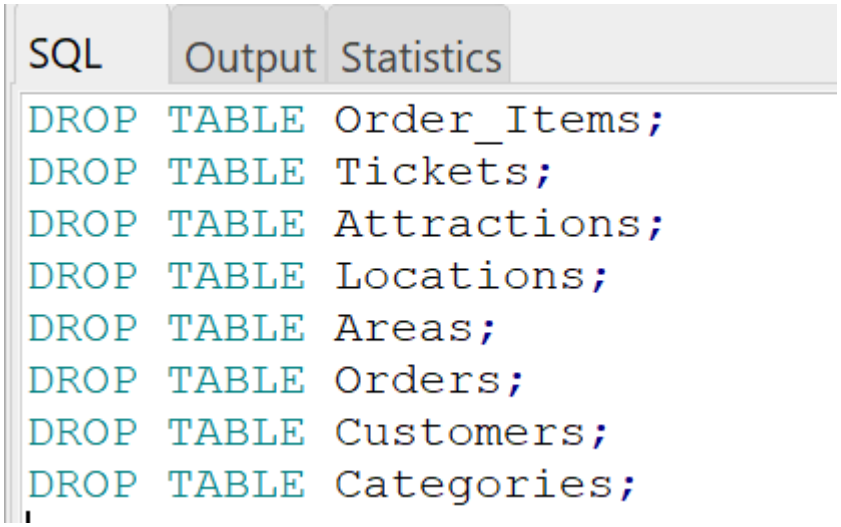
```
CREATE TABLE Customers
(
 customer_id INT NOT NULL,
 customer_name INT NOT NULL,
 phone_numer INT NOT NULL,
 address_line_1 INT NOT NULL,
 address_line_2 INT,
 PRIMARY KEY (customer_id)
);

CREATE TABLE Orders
(
 order_date INT NOT NULL,
 order_id INT NOT NULL,
 customer_id INT NOT NULL,
 PRIMARY KEY (order_id),
 FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

CREATE TABLE Areas
(
 area_id INT NOT NULL,
 area_name INT NOT NULL,
 PRIMARY KEY (area_id)
);

CREATE TABLE Locations
(
 location_id INT NOT NULL,
 location_name INT NOT NULL,
 area_id INT NOT NULL,
 PRIMARY KEY (location_id),
 FOREIGN KEY (area_id) REFERENCES Areas(area_id)
);

CREATE TABLE Attractions
(
 attraction_id INT NOT NULL,
 attraction_name INT NOT NULL,
 Description INT NOT NULL,
 opening_hours INT NOT NULL,
 location_id INT NOT NULL,
 PRIMARY KEY (attraction_id),
 FOREIGN KEY (location_id) REFERENCES Locations(location_id)
);

CREATE TABLE Tickets
(
 ticket_id INT NOT NULL,
 price INT NOT NULL,
 valid_from INT NOT NULL,
 valid_until INT NOT NULL,
 category_id INT NOT NULL,
 attraction_id INT NOT NULL,
 PRIMARY KEY (ticket_id),
```

```
 FOREIGN KEY (category_id) REFERENCES Categories(category_id),
 FOREIGN KEY (attraction_id) REFERENCES Attractions(attraction_id)
);

CREATE TABLE Order_Items
(
 ticket_id INT NOT NULL,
 order_id INT NOT NULL,
 PRIMARY KEY (ticket_id, order_id),
 FOREIGN KEY (ticket_id) REFERENCES Tickets(ticket_id),
 FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);
```

Mapping:

Entity	Attribute	Description
Categories	category_id	Unique identifier for each category.
	category_name	Name of the category (e.g., "Adult," "Teenager").
	min_age	Minimum age requirement for the category.
Customers	customer_id	Unique identifier for each customer.
	customer_name	Name of the customer.
	phone_numer	Phone number of the customer.
	address_line_1	First line of the customer's address.
	address_line_2	Second line of the customer's address (optional).
Orders	order_date	Date when the order was placed.
	order_id	Unique identifier for each order.
	customer_id	Foreign key referencing the customer who placed the order.
Areas	area_id	Unique identifier for each area.
	area_name	Name of the area (e.g., "North," "South").
Locations	location_id	Unique identifier for each location.
	location_name	Name of the location.
	area_id	Foreign key referencing the area to which the location belongs.
Attractions	attraction_id	Unique identifier for each attraction.
	attraction_name	Name of the attraction.
	Description	Description of the attraction.
	opening_hours	Opening hours of the attraction.
	location_id	Foreign key referencing the location where the attraction is located.
Tickets	ticket_id	Unique identifier for each ticket.
	price	Price of the ticket.
	valid_from	Validity start date of the ticket.
	valid_until	Validity end date of the ticket.
	category_id	Foreign key referencing the category of the ticket.
	attraction_id	Foreign key referencing the attraction for which the ticket is valid.
Order_Items	ticket_id	Foreign key referencing the ticket included in the order item.
	order_id	Foreign key referencing the order to which the order item belongs.

Drop Tables:



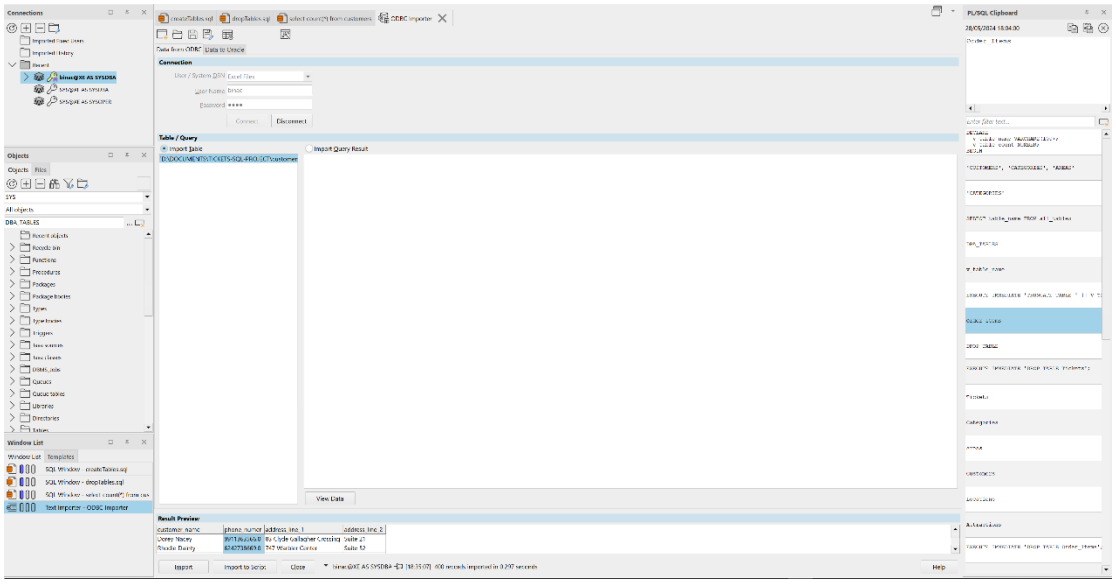
Insert table command:

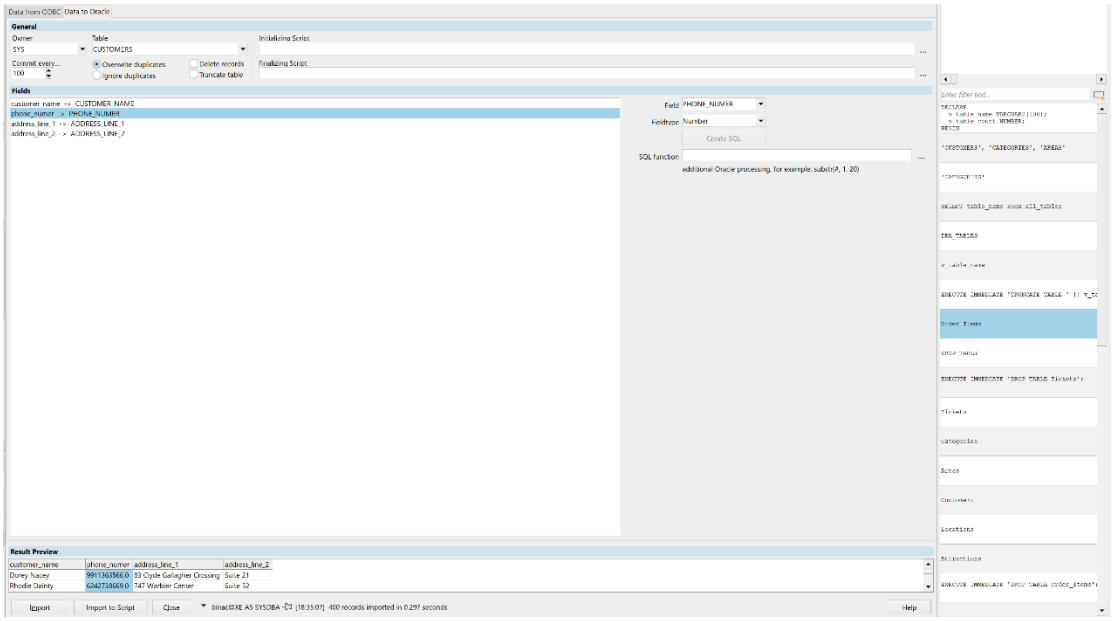
SQL Output Statistics

```
INSERT INTO Categories (category_name, min_age) VALUES ('Adult', 18);
INSERT INTO Categories (category_name, min_age) VALUES ('Teenager', 13);
INSERT INTO Categories (category_name, min_age) VALUES ('Child', 0);
```

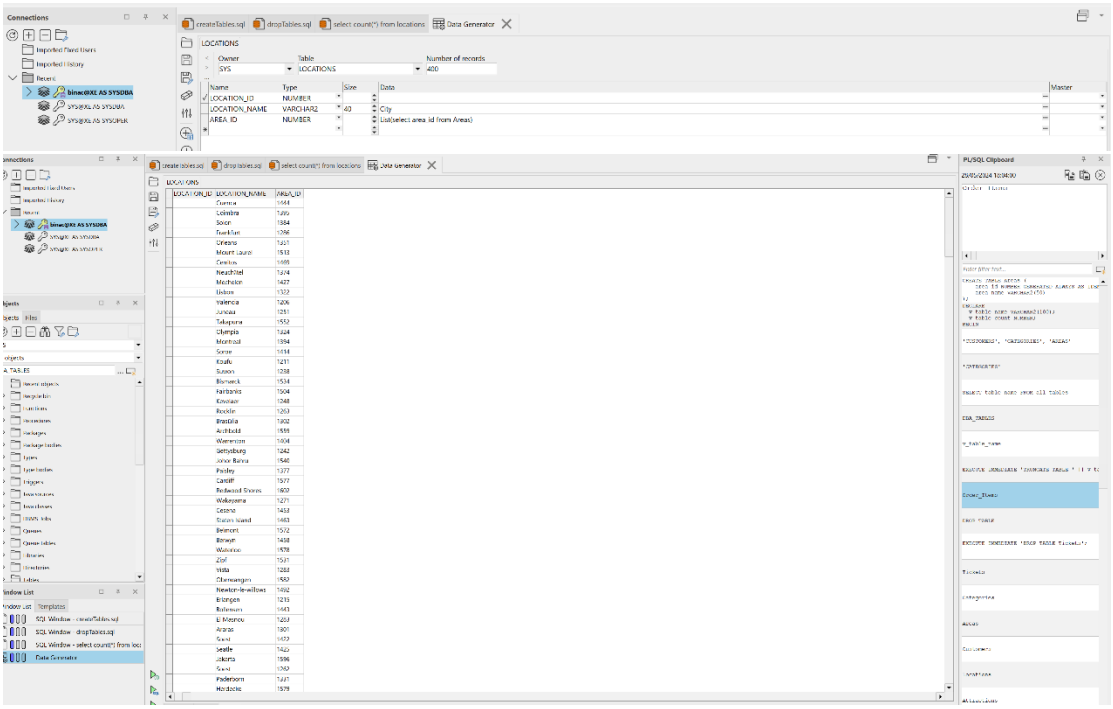
Select All:

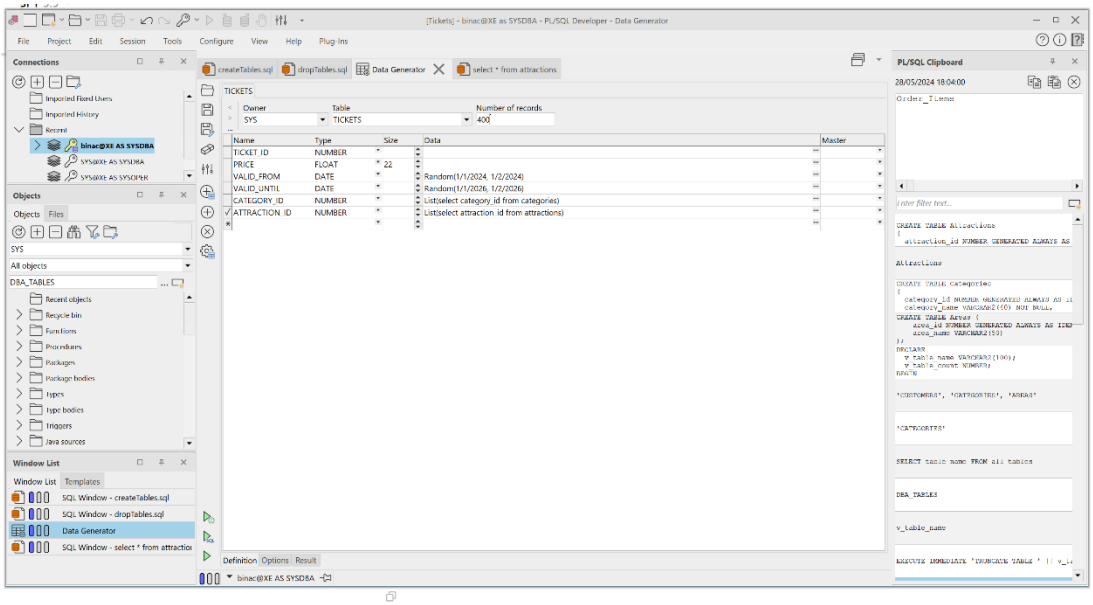
Import from Excel file:

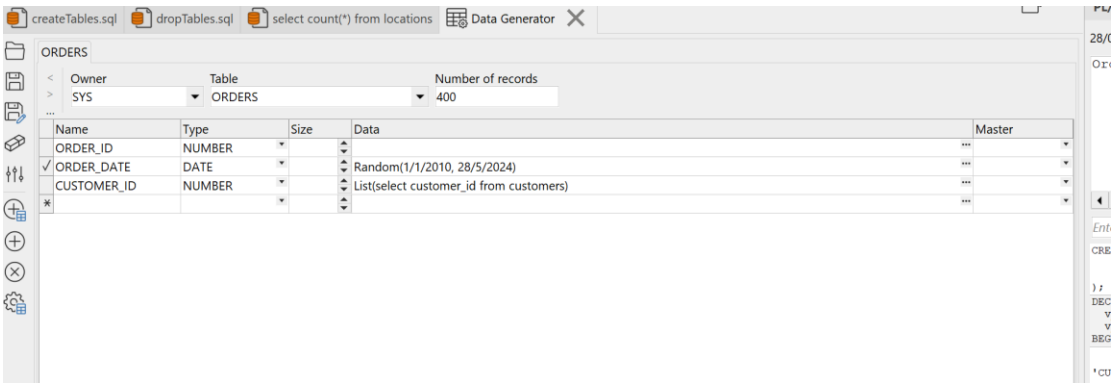




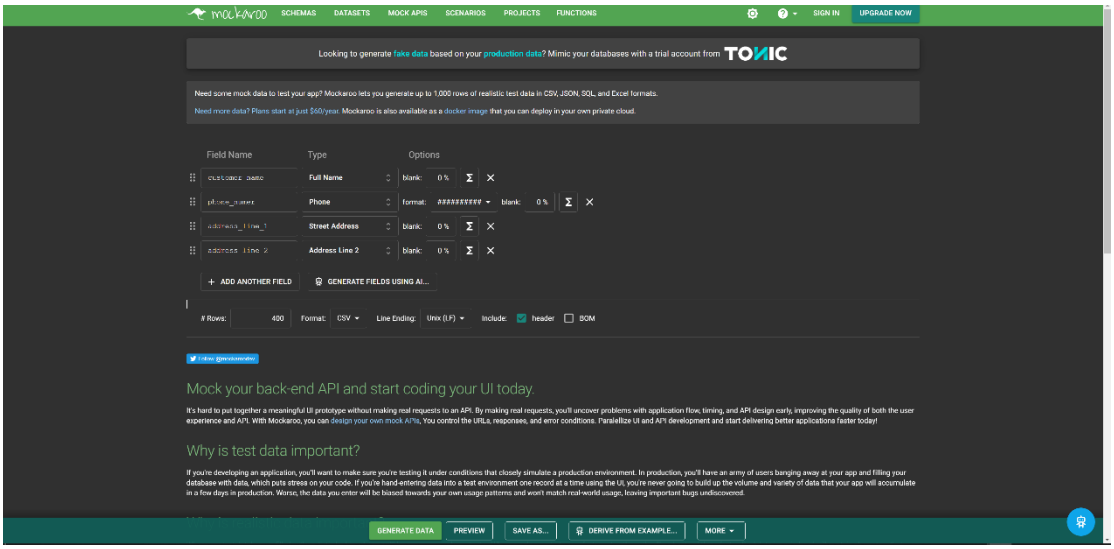
Data Generator:

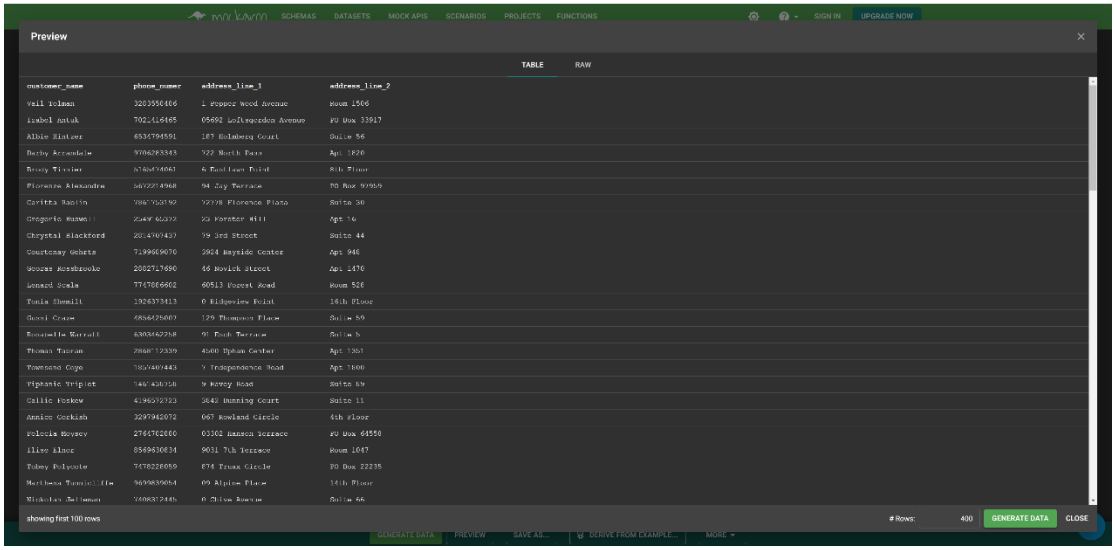



[Tickets] - binac@XE as SYSDBA - PL/SQL Developer - Data Generator
createTables.sql
dropTables.sql
Data Generator
select * from attractions
TICKETS
Owner
SYS
Table
TICKETS
Number of records
400
Name
Type
Size
Data
Master
TICKET_ID
NUMBER
PRICE
FLOAT
22
VALID_FROM
DATE
Random(1/1/2024, 1/2/2024)
VALID_UNTIL
DATE
Random(1/1/2026, 1/2/2026)
CATEGORY_ID
NUMBER
List(select category_id from categories)
ATTRACTION_ID
NUMBER
List(select attraction_id from attractions)
PL/SQL Clipboard
28/05/2024 18:04:00

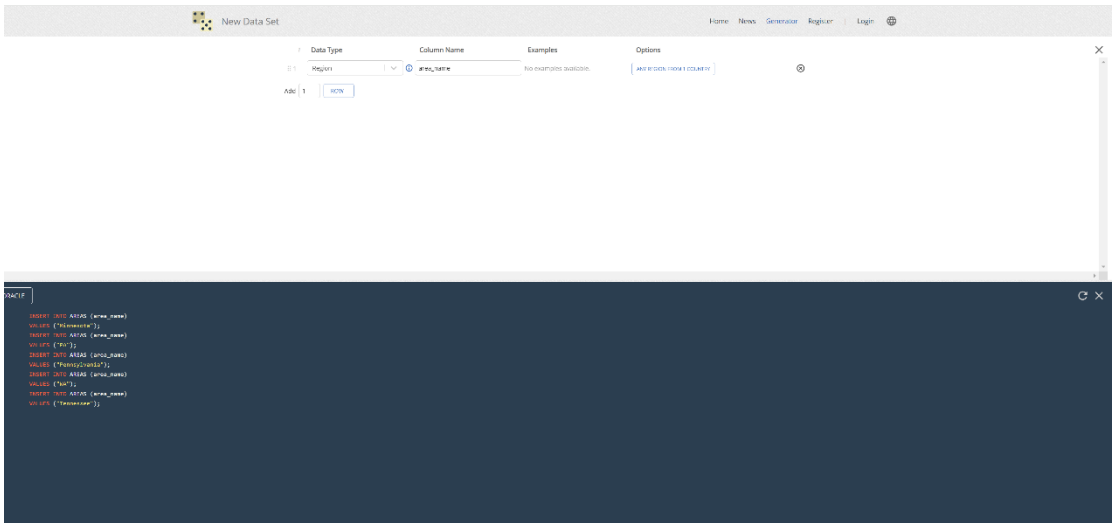

createTables.sql
dropTables.sql
select count(*) from locations
Data Generator
ORDERS
Owner
SYS
Table
ORDERS
Number of records
400
Name
Type
Size
Data
Master
ORDER_ID
NUMBER
ORDER_DATE
DATE
Random(1/1/2010, 28/5/2024)
CUSTOMER_ID
NUMBER
List(select customer_id from customers)

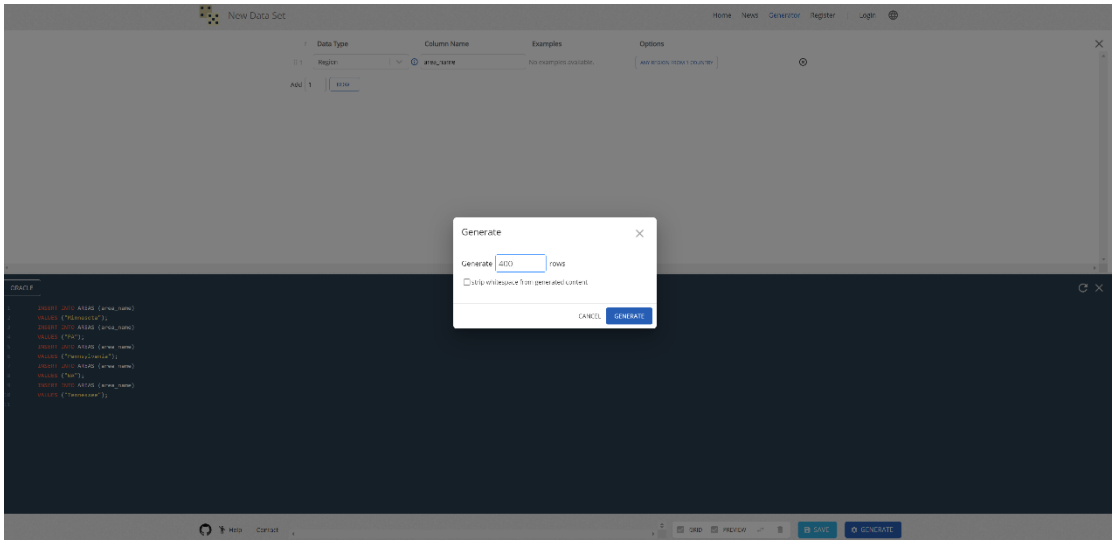
Mockaroo:


mockaroo
SCHEMAS
DATASETS
MOCK APIS
SCENARIOS
PROJECTS
FUNCTIONS
SIGN IN
UPGRADE NOW
Looking to generate fake data based on your production data? Mimic your databases with a trial account from TONIC
Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.
Need more data? Plans start at just $60/year. Mockaroo is also available as a docker image that you can deploy in your own private cloud.
Field Name
Type
Options
customer_name
Full Name
phone_number
Phone
address_line_1
Street Address
address_line_2
Address Line 2
ADD ANOTHER FIELD
GENERATE FIELDS USING AI...
Rows:
400
Format:
CSV
Line Ending:
Unix (LF)
Include:
header
BOM
Mock your back-end API and start coding your UI today.
Why is test data important?
GENERATE DATA
PREVIEW
SAVE AS...
DERIVE FROM EXAMPLE...
MORE



Generate data:





SQL Output Statistics

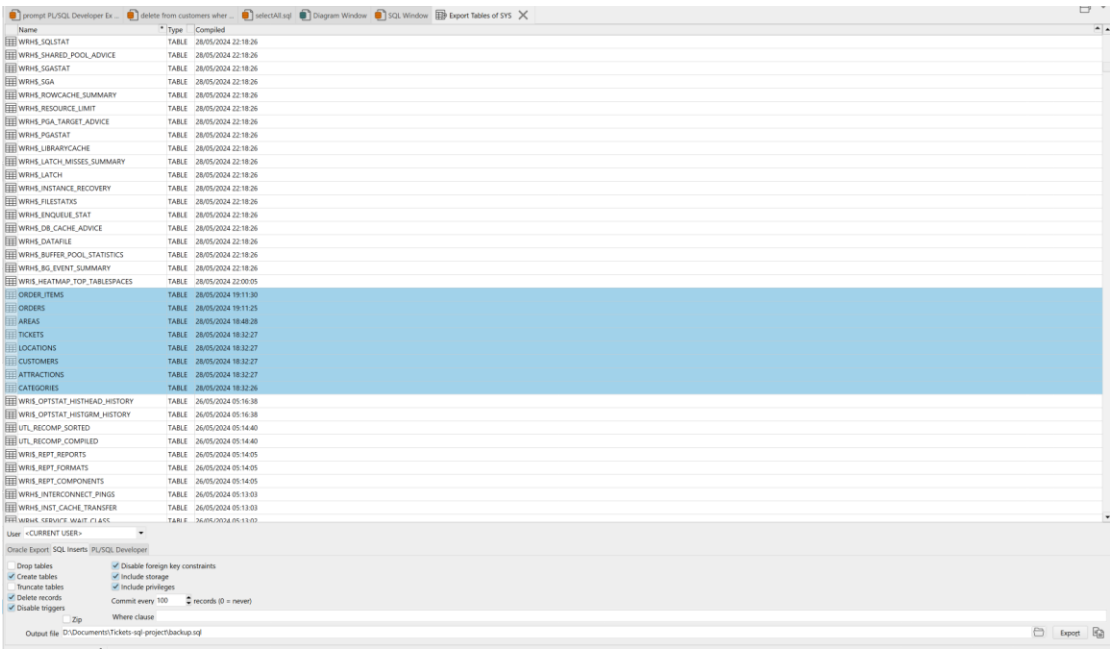
```
select * from customers;
select * from orders;
select * from order_items;
select * from tickets;
select * from locations;
select * from areas;
select * from attractions;
select * from categories;
```

Select customers | Select orders | Select order_items | Select tickets | Select locations | Select areas | Select attractions | Select categories

	CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMER	ADDRESS_LINE_1	ADDRESS_LINE_2
1	1	Dorey Nacey	9911363566	83 Clyde Gallagher Crossing	Suite 21
2	2	Rhodie Dainty	6242738669	747 Warbler Center	Suite 52
3	3	Jobey Abrahm	4734466220	21701 Sheridan Court	Suite 6
4	4	Ryley Trobridge	2785068828	05 Hayes Center	Apt 1357
5	5	Elbertina Farrah	6706256225	26 Monica Lane	Suite 27
6	6	Vale McNess	2698846507	377 Independence Road	Apt 1947
7	7	Oralla Ethridge	5101711356	9 Atwood Street	Apt 311
8	8	Sarah Caen	2784782898	862 Bluestem Lane	Room 1429
9	9	Jana Beglin	2674184360	5260 Badeau Park	7th Floor
10	10	Douglass Thorington	3439944110	1022 Village Place	PO Box 33399
11	11	Ricardo Carhart	4917941819	116 Stuart Way	PO Box 47836
12	12	Charlotte Hazle	4803922669	8159 Farmco Crossing	12th Floor
13	13	Kelley Itzcovich	1482199105	629 Gateway Lane	Room 162
14	14	Bronnie Ryce	6028558674	2493 Springview Lane	Apt 454
15	15	Adolf Walbridge	2121598913	585 East Point	1st Floor
16	16	Dario Tythe	4766066493	08 Londonderry Drive	Apt 1069
17	17	Jocelyne Wemes	6035333588	533 Anniversary Trail	Room 1269
18	18	Renae Dybbe	3632020470	4836 Gina Lane	Suite 5
19	19	Perceval Yarranton	3771513579	285 Prairieview Court	PO Box 29149
20	20	Riannon Posnett	2995383967	312 Elgar Point	Suite 14
21	21	Virgil MacCurley	7654257109	248 Arkansas Plaza	Room 14
22	22	Madelle Comerford	5211416319	84428 Fordem Alley	Room 997
23	23	Ty Ineson	7556760927	0 Warner Terrace	PO Box 41711
24	24	Rycca Alejo	9549349862	1191 Packers Alley	PO Box 99923
25	25	Petronilla Blethyn	4455948399	92773 Dovetail Place	Suite 32

```
SQL> desc areas
Name      Type         Nullable Default                       Comments
--------- ------------ -------- ----------------------------- --------
AREA_ID   NUMBER                "SYS"."ISEQ$$_77372".nextval
AREA_NAME VARCHAR2(50) Y

SQL> desc locations
Name          Type         Nullable Default                       Comments
------------- ------------ -------- ----------------------------- --------
LOCATION_ID   NUMBER                "SYS"."ISEQ$$_77375".nextval
LOCATION_NAME VARCHAR2(40)
AREA_ID       INTEGER

SQL> desc attractions
Name            Type          Nullable Default                       Comments
--------------- ------------- -------- ----------------------------- --------
ATTRACTION_ID   NUMBER                  "SYS"."ISEQ$$_77378".nextval
ATTRACTION_NAME VARCHAR2(40)
DESCRIPTION     VARCHAR2(100)
OPENING_HOURS   VARCHAR2(40)
LOCATION_ID     INTEGER

SQL> desc categories
Name          Type         Nullable Default                       Comments
------------- ------------ -------- ----------------------------- --------
CATEGORY_ID   NUMBER                "SYS"."ISEQ$$_77363".nextval
CATEGORY_NAME VARCHAR2(40)
MIN_AGE       INTEGER

SQL> desc tickets
Name          Type    Nullable Default                       Comments
------------- ------- -------- ----------------------------- --------
TICKET_ID     NUMBER           "SYS"."ISEQ$$_77381".nextval
PRICE         FLOAT
VALID_FROM    DATE
VALID_UNTIL   DATE
CATEGORY_ID   INTEGER
ATTRACTION_ID INTEGER

SQL> desc customers
Name           Type         Nullable Default                       Comments
-------------- ------------ -------- ----------------------------- --------
CUSTOMER_ID    NUMBER                "SYS"."ISEQ$$_77366".nextval
CUSTOMER_NAME  VARCHAR2(40)
PHONE_NUMER    NUMBER(10)
ADDRESS_LINE_1 VARCHAR2(40)
ADDRESS_LINE_2 VARCHAR2(40) Y

SQL> desc orders
Name        Type    Nullable Default                       Comments
----------- ------- -------- ----------------------------- --------
ORDER_ID    NUMBER           "SYS"."ISEQ$$_77387".nextval
ORDER_DATE  DATE
CUSTOMER_ID INTEGER

SQL> desc order_items
Name      Type    Nullable Default Comments
--------- ------- -------- ------- --------
TICKET_ID INTEGER
ORDER_ID  INTEGER
```

Backup:


prompt PL/SQL Developer Ex ...
delete from customers wher ...
selectAll.sql
Diagram Window
SQL Window
Export Tables of SYS
Name
Type
Compiled
WRH$_SQLSTAT TABLE 28/05/2024 22:18:26
WRH$_SHARED_POOL_ADVICE TABLE 28/05/2024 22:18:26
WRH$_SGASTAT TABLE 28/05/2024 22:18:26
WRH$_SGA TABLE 28/05/2024 22:18:26
WRH$_ROWCACHE_SUMMARY TABLE 28/05/2024 22:18:26
WRH$_RESOURCE_LIMIT TABLE 28/05/2024 22:18:26
WRH$_PGA_TARGET_ADVICE TABLE 28/05/2024 22:18:26
WRH$_PGASTAT TABLE 28/05/2024 22:18:26
WRH$_LIBRARYCACHE TABLE 28/05/2024 22:18:26
WRH$_LATCH_MISSES_SUMMARY TABLE 28/05/2024 22:18:26
WRH$_LATCH TABLE 28/05/2024 22:18:26
WRH$_INSTANCE_RECOVERY TABLE 28/05/2024 22:18:26
WRH$_FILESTATXS TABLE 28/05/2024 22:18:26
WRH$_ENQUEUE_STAT TABLE 28/05/2024 22:18:26
WRH$_DB_CACHE_ADVICE TABLE 28/05/2024 22:18:26
WRH$_DATAFILE TABLE 28/05/2024 22:18:26
WRH$_BUFFER_POOL_STATISTICS TABLE 28/05/2024 22:18:26
WRH$_BG_EVENT_SUMMARY TABLE 28/05/2024 22:18:26
WRI$_HEATMAP_TOP_TABLESPACES TABLE 28/05/2024 22:00:05
ORDER_ITEMS TABLE 28/05/2024 19:11:30
ORDERS TABLE 28/05/2024 19:11:25
AREAS TABLE 28/05/2024 18:48:28
TICKETS TABLE 28/05/2024 18:32:27
LOCATIONS TABLE 28/05/2024 18:32:27
CUSTOMERS TABLE 28/05/2024 18:32:27
ATTRACTIONS TABLE 28/05/2024 18:32:27
CATEGORIES TABLE 28/05/2024 18:32:26
WRI$_OPTSTAT_HISTHEAD_HISTORY TABLE 26/05/2024 05:16:38
WRI$_OPTSTAT_HISTGRM_HISTORY TABLE 26/05/2024 05:16:38
UTL_RECOMP_SORTED TABLE 26/05/2024 05:14:40
UTL_RECOMP_COMPILED TABLE 26/05/2024 05:14:40
WRI$_REPT_REPORTS TABLE 26/05/2024 05:14:05
WRI$_REPT_FORMATS TABLE 26/05/2024 05:14:05
WRI$_REPT_COMPONENTS TABLE 26/05/2024 05:14:05
WRH$_INTERCONNECT_PINGS TABLE 26/05/2024 05:13:03
WRH$_INST_CACHE_TRANSFER TABLE 26/05/2024 05:13:03
User <CURRENT USER>
Oracle Export
SQL Inserts
PL/SQL Developer
Drop tables
Create tables
Truncate tables
Delete records
Disable triggers
Zip
Disable foreign key constraints
Include storage
Include privileges
Commit every 100 records (0 = never)
Where clause
Output file D:\Documents\Tickets-sql-project\backup.sql
Export