

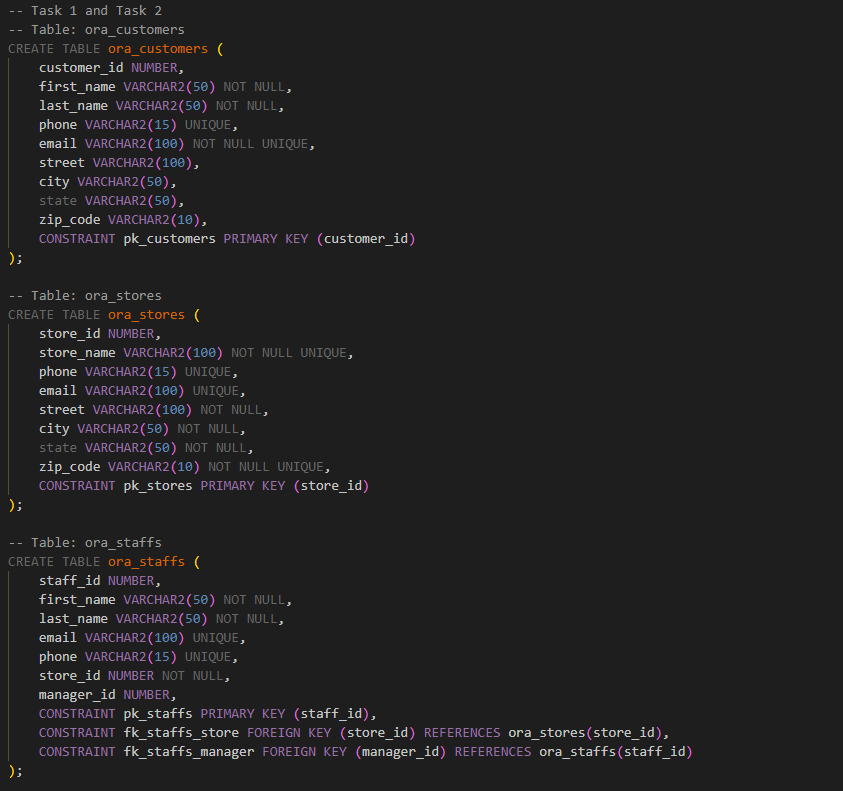
| Submitted By: | Kriston Jomari (231165R) |
| --- | --- |
|  | Wong Jun Lin (242844E) |
|  | Lim Wei Yi, Markus (240785A) |
|  | Kiew Hein Royce (231003P) |
| Module Group: | IT1393-02 |
| Tutor: | Ms. Sally Tang |
| Submitted on: | 08/12/2024 |

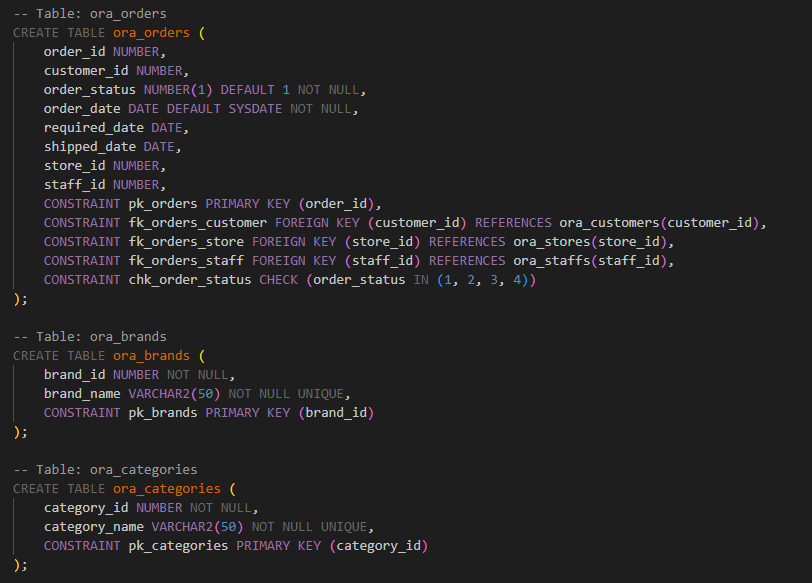


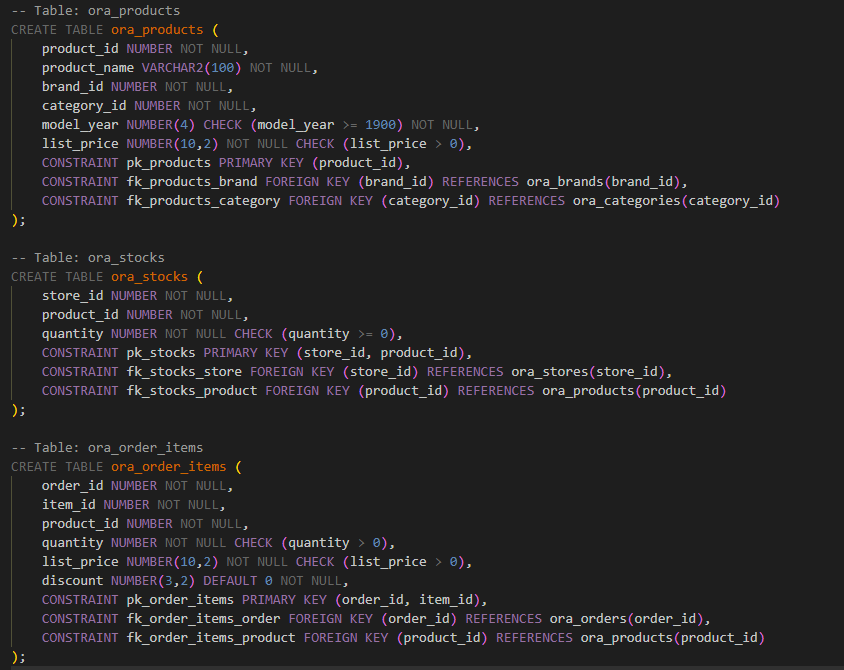
# Part 1: Defining Database Structure and Objects using Database Definition Language (DDL) for KeithsBike database (50 marks)

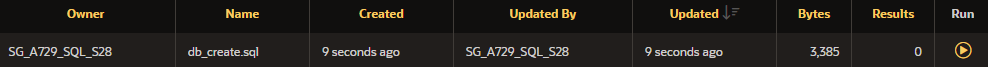
**Task 1: Develop DDL statements for creating the tables in the KeithsBike Database**

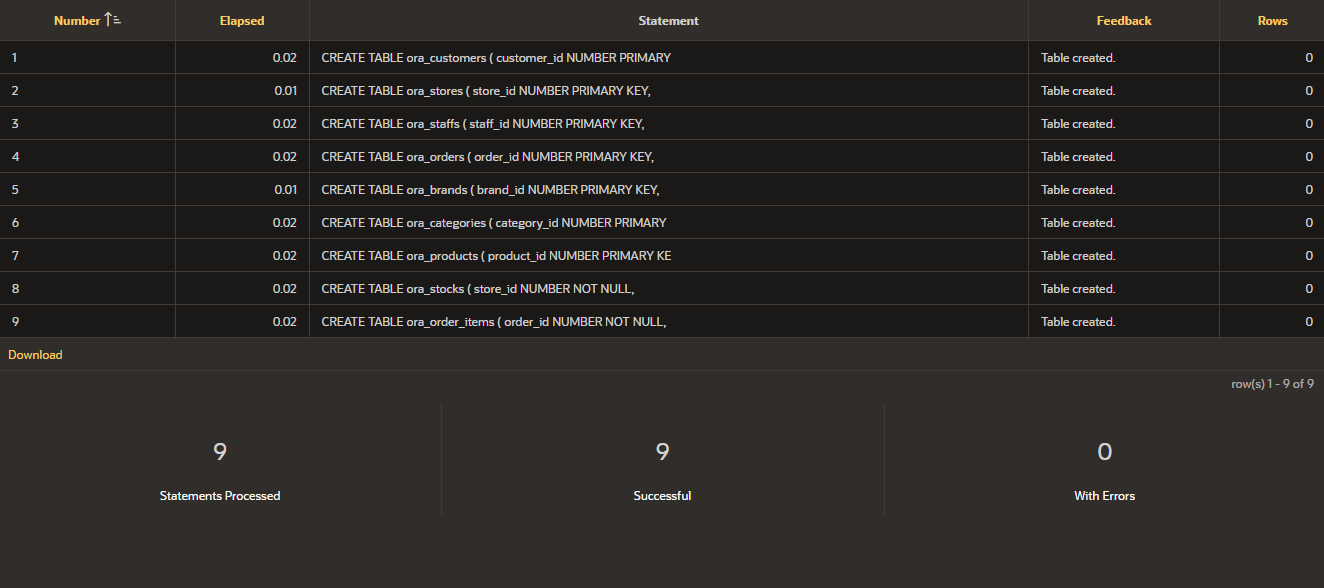
The initial DDL statements used.

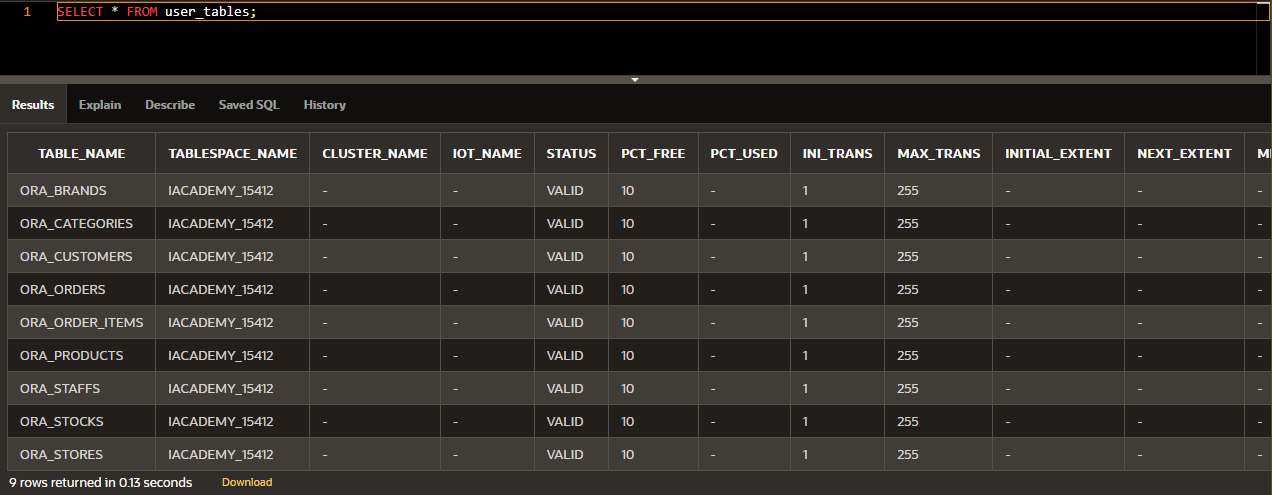






In Oracle APEX, visit SQL Workshop > SQL Scripts > Upload > Run > Run Now

Proof showing that all nine statements were successfully created:  


Testing with SELECT \* FROM user\_tables; provides us a list of all nine tables.  


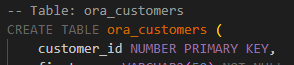
**Assumptions Made:**

1. **The stores are located in the**

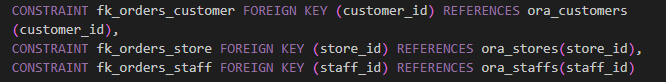
**Task 2: Creating and Managing Constraints**

1. Adding the different constraints.  
   1. Primary Key (PK) and Foreign Key (FK) constraints

All relevant PKs and FKs were predefined in the script in Task 1. Below are the relevant snippets of code:

ora\_customers  


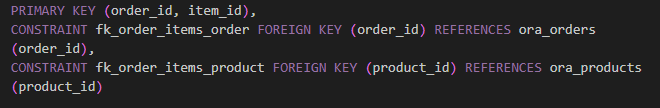
ora\_orders  

ora\_staffs  

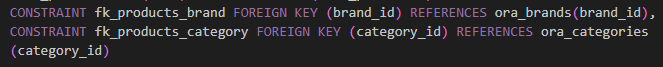


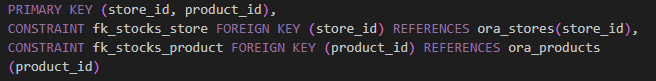

ora\_stores  


ora\_order\_items  


ora\_categories  


ora\_products  

ora\_stocks  


ora\_brands  


* 1. Unique (UQ) and Not Null (NN) columns with explanation (might not be accurate as changes made)

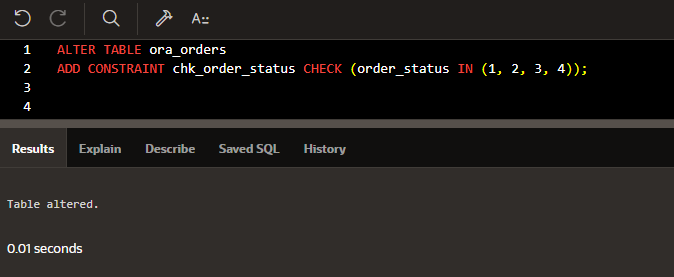
| Table | Column | UQ | NULL Allowed | Explanation |
| --- | --- | --- | --- | --- |
| **ora\_customers** | customer\_id | ✔ | ✘ | Each customer must have a unique ID (Primary Key). |
|  | first\_name | ✘ | ✘ | Every customer needs a first name. |
|  | last\_name | ✘ | ✘ | Every customer needs a last name. |
|  | phone | ✔ | ✔ | Phone numbers are optional but must be unique. |
|  | email | ✔ | ✔ | Emails are required and must not repeat. |
|  | street | ✘ | ✔ | Address is optional. |
|  | city | ✘ | ✔ | City is optional. |
|  | state | ✘ | ✔ | State is optional. |
|  | zip\_code | ✘ | ✔ | Zip code is optional. |
| **ora\_orders** | order\_id | ✔ | ✘ | Each order must have a unique ID (Primary Key). |
|  | customer\_id | ✘ | ✘ | Every order must belong to a customer. |
|  | order\_status | ✘ | ✘ | Status is required and defaults to "1" (Pending). |
|  | order\_date | ✘ | ✘ | Defaults to today if not explicitly given. |
|  | required\_date | ✘ | ✔ | Required delivery date is optional. |
|  | shipped\_date | ✘ | ✔ | Shipping date is optional until shipped. |
|  | store\_id | ✘ | ✘ | Orders must come from a store. |
|  | staff\_id | ✘ | ✔ | Staff ID is optional if no staff is assigned. |
| **ora\_staffs** | staff\_id | ✔ | ✘ | Each staff member must have a unique ID (Primary Key). |
|  | first\_name | ✘ | ✘ | Every staff member needs a first name. |
|  | last\_name | ✘ | ✘ | Every staff member needs a last name. |
|  | email | ✔ | ✔ | Email is optional but must be unique if provided. |
|  | phone | ✔ | ✔ | Phone is optional but must be unique if provided. |
|  | store\_id | ✘ | ✘ | Every staff member must belong to a store. |
|  | manager\_id | ✘ | ✔ | Manager is optional for top-level staff. |
| **ora\_stores** | store\_id | ✔ | ✘ | Each store must have a unique ID. |
|  | store\_name | ✘ | ✘ | Store name is required. |
|  | phone | ✔ | ✔ | Phone is optional but must be unique if provided. |
|  | email | ✘ | ✔ | Email is optional. |
|  | street | ✘ | ✘ | Address is required. |
|  | city | ✘ | ✘ | City is required. |
|  | state | ✘ | ✘ | State is required. |
|  | zip\_code | ✔ | ✘ | Zip code is unique and is required. |
| **ora\_products** | product\_id | ✔ | ✘ | Each product must have a unique ID (Primary Key). |
|  | product\_name | ✘ | ✘ | Every product needs a name. |
|  | brand\_id | ✘ | ✘ | Every product must belong to a brand. |
|  | category\_id | ✘ | ✘ | Every product must belong to a category. |
|  | model\_year | ✘ | ✔ | Year is optional. |
|  | list\_price | ✘ | ✘ | Price is required and must be greater than 0. |
| **ora\_categories** | category\_id | ✔ | ✘ | Each category must have a unique ID. |
|  | category\_name | ✔ | ✘ | Every category needs a unique name. |
| **ora\_brands** | brand\_id | ✔ | ✘ | Each brand must have a unique ID. |
|  | brand\_name | ✔ | ✘ | Every brand needs a unique name. |
| **ora\_order\_items** | order\_id | ✘ | ✘ | Must reference an existing order (Part of PK). |
|  | item\_id | ✘ | ✘ | Unique within each order (Part of PK). |
|  | product\_id | ✘ | ✘ | Every order item must have a product ID. |
|  | quantity | ✘ | ✘ | Quantity is required and must be more than 0. |
|  | list\_price | ✘ | ✘ | Price is required and must be greater than 0. |
|  | discount | ✘ | ✔ | Discount is optional and defaults to 0. |
| **ora\_stocks** | store\_id | ✘ | ✘ | Must reference an existing store (Part of PK). |
|  | product\_id | ✘ | ✘ | Must reference an existing product (Part of PK). |
|  | quantity | ✘ | ✘ | Quantity is required and must be at least 0. |

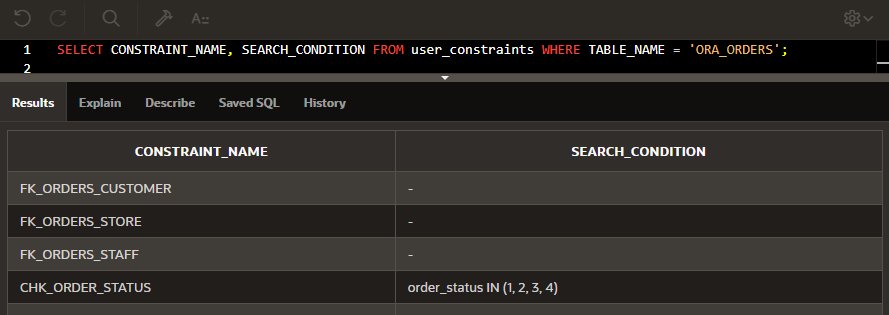
* 1. CHECK constraint to ensure price is > $0.00 (also implemented in Task 1).

CHECK constraint on list\_price in ora\_products.

list\_price NUMBER(10,2) NOT NULL CHECK (list\_price > 0)

* 1. CHECK constraint to restrict order\_status values.

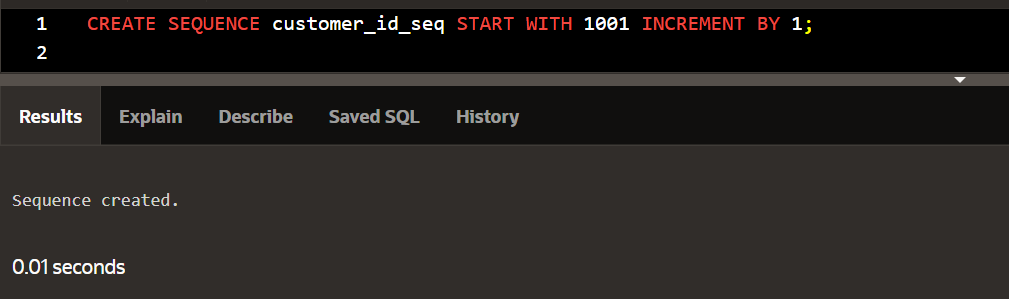
CHECK constraint on order\_status in ora\_orders.

For checking.  


**Task 3: Working with Sequences (Indexes and Synonyms) & View**

1. Use a sequence to generate PKs for the Customers table beginning from 1001 and increment by 1. We used:

CREATE SEQUENCE customer\_id\_seq START WITH 1001 INCREMENT BY 1;



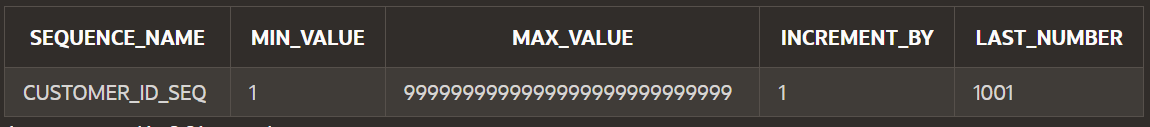
Check with:

SELECT SEQUENCE\_NAME, MIN\_VALUE, MAX\_VALUE, INCREMENT\_BY, LAST\_NUMBER

FROM user\_sequences

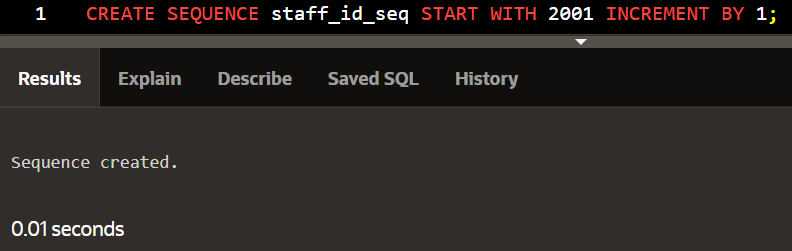
WHERE SEQUENCE\_NAME = 'CUSTOMER\_ID\_SEQ';

Results:



1. Use a sequence to generate PKs for Staffs table starting at 2001 and increment by 1. We used:

CREATE SEQUENCE staff\_id\_seq START WITH 2001 INCREMENT BY 1;



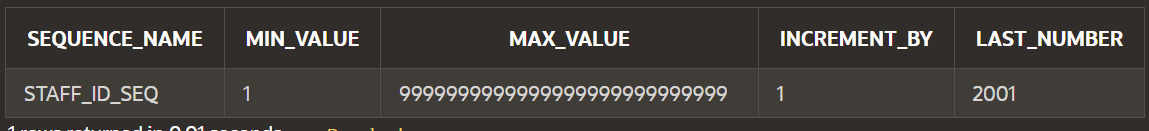
Check with:

SELECT SEQUENCE\_NAME, MIN\_VALUE, MAX\_VALUE, INCREMENT\_BY, LAST\_NUMBER

FROM user\_sequences

WHERE SEQUENCE\_NAME = 'STAFF\_ID\_SEQ';

Results



1. Add the data to the tables using INSERT statements.

Brands:

-- Insert records into ora\_brands

INSERT INTO ora\_brands (brand\_id, brand\_name)

VALUES (1, 'Electra');

INSERT INTO ora\_brands (brand\_id, brand\_name)

VALUES (2, 'Haro');

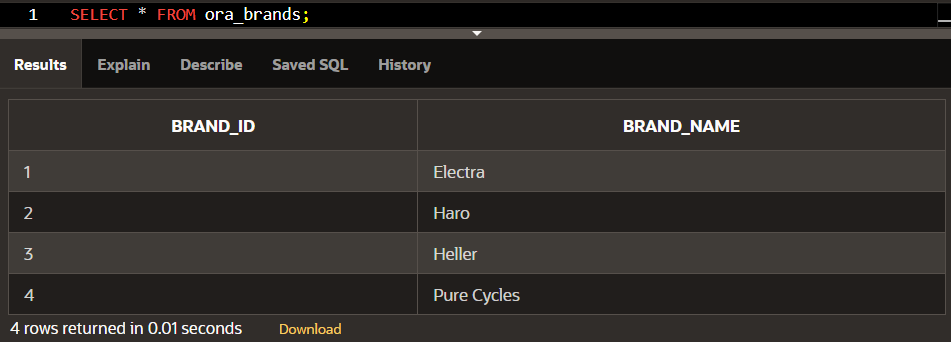
INSERT INTO ora\_brands (brand\_id, brand\_name)

VALUES (3, 'Heller');

INSERT INTO ora\_brands (brand\_id, brand\_name)

VALUES (4, 'Pure Cycles');

Proof:



Stores:

-- Insert records into ora\_stores

INSERT INTO ora\_stores (store\_id, store\_name, phone, email, street, city, state, zip\_code)

VALUES (111, 'Santa Bikes', '831-476-4321', 'santa@Keithsbikes.com', '3700 Portola Drive', 'Santa Cruz', 'CA', '95060');

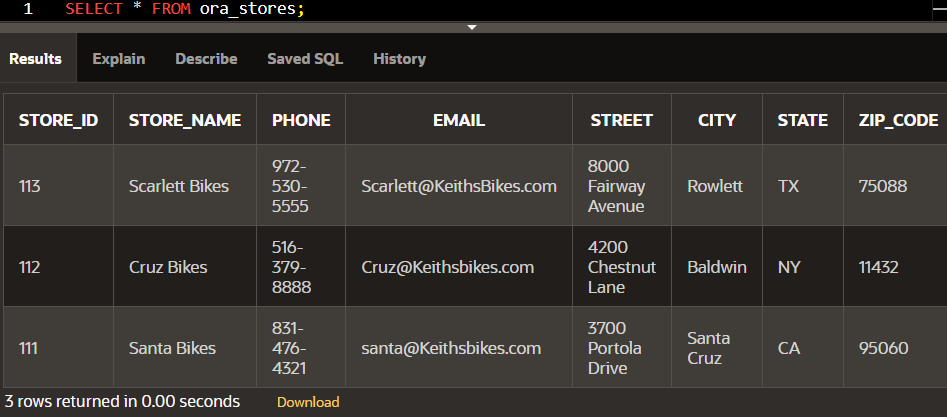
INSERT INTO ora\_stores (store\_id, store\_name, phone, email, street, city, state, zip\_code)

VALUES (112, 'Cruz Bikes', '516-379-8888', 'Cruz@Keithsbikes.com', '4200 Chestnut Lane', 'Baldwin', 'NY', '11432');

INSERT INTO ora\_stores (store\_id, store\_name, phone, email, street, city, state, zip\_code)

VALUES (113, 'Scarlett Bikes', '972-530-5555', 'Scarlett@KeithsBikes.com', '8000 Fairway Avenue', 'Rowlett', 'TX', '75088');

Proof:



Categories:

-- Insert records into ora\_categories

INSERT INTO ora\_categories (category\_id, category\_name)

VALUES (11, 'Children Bicycles');

INSERT INTO ora\_categories (category\_id, category\_name)

VALUES (12, 'Electric Bikes');

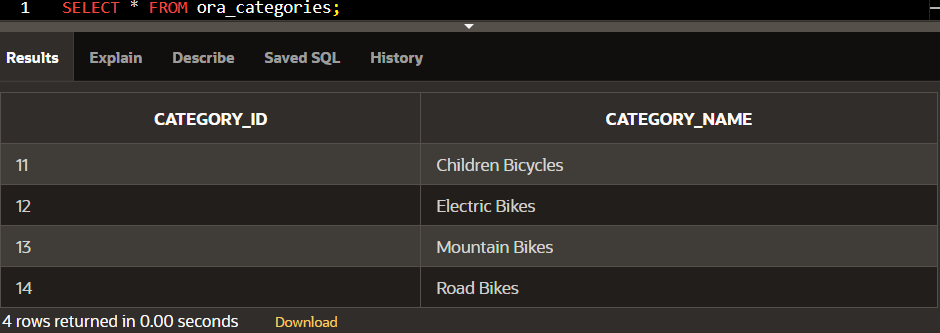
INSERT INTO ora\_categories (category\_id, category\_name)

VALUES (13, 'Mountain Bikes');

INSERT INTO ora\_categories (category\_id, category\_name)

VALUES (14, 'Road Bikes');

Proof:



Products:

-- Insert records into ora\_products

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (1, 'Trek 820 - 2016', 2, 13, 2016, 379);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (2, 'Ritchey Timberwolf Frameset - 2016', 2, 13, 2016, 749.99);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (3, 'Surly Wednesday Frameset - 2016', 3, 13, 2016, 899);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (6, 'Sun Bicycles Streamway - 2017', 2, 12, 2017, 480);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (7, 'Sun Bicycles Cruz 7 - 2017', 4, 11, 2017, 420);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (8, 'Sun Bicycles Drifter 7 - Women''s - 2017', 4, 11, 2017, 479.99);

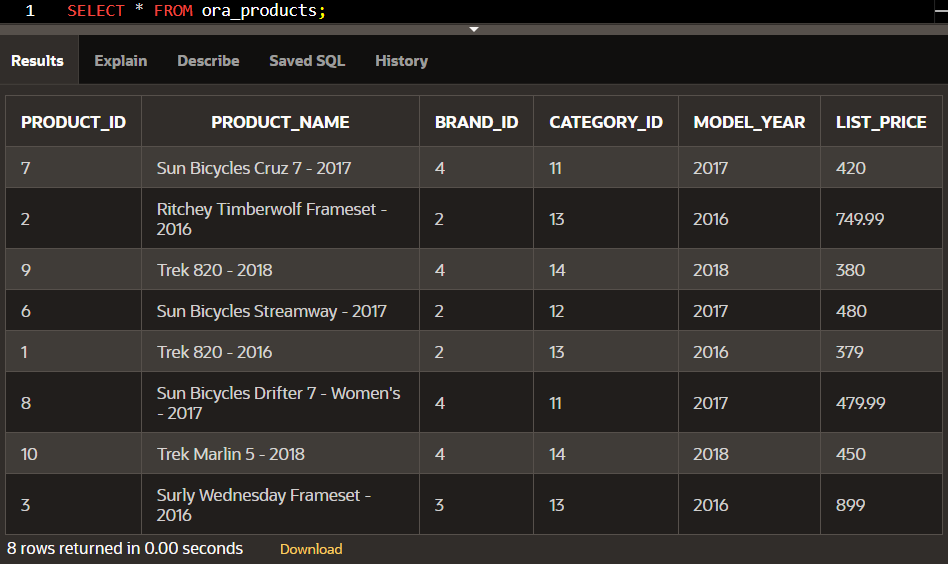
INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (9, 'Trek 820 - 2018', 4, 14, 2018, 380);

INSERT INTO ora\_products (product\_id, product\_name, brand\_id, category\_id, model\_year, list\_price)

VALUES (10, 'Trek Marlin 5 - 2018', 4, 14, 2018, 450);

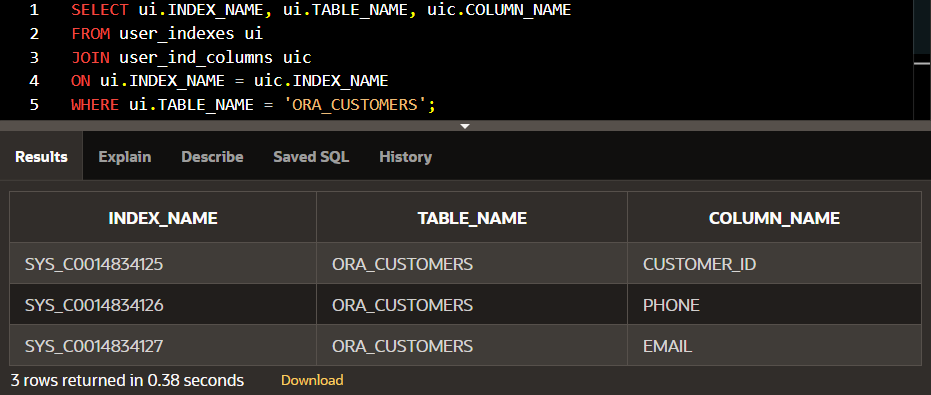
Proof:



1. Add an index on the email column for the Customers table. We used:

CREATE INDEX idx\_customer\_email ON ora\_customers (email);

Proof:



1. Create a view. We used:

CREATE VIEW vw\_staff\_store\_orders AS

SELECT

s.first\_name || ' ' || s.last\_name AS staff\_name,

st.store\_name,

o.order\_id

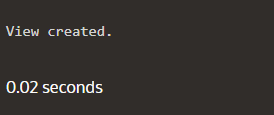
FROM ora\_staffs s

JOIN ora\_orders o ON s.staff\_id = o.staff\_id

JOIN ora\_stores st ON s.store\_id = st.store\_id

WHERE o.order\_status != 4 -- Exclude completed orders

WITH READ ONLY;



For the view to show data, we first have to have data. Below are the sample data that was inserted:

For ora\_staffs:

INSERT INTO ora\_staffs (staff\_id, first\_name, last\_name, email, phone, store\_id)

VALUES (2001, 'Alice', 'Johnson', 'alice.johnson@santabikes.com', '123-456-7890', 111);

INSERT INTO ora\_staffs (staff\_id, first\_name, last\_name, email, phone, store\_id)

VALUES (2002, 'Bob', 'Williams', 'bob.williams@cruzbikes.com', '987-654-3210', 112);

INSERT INTO ora\_staffs (staff\_id, first\_name, last\_name, email, phone, store\_id)

VALUES (2003, 'Claire', 'Smith', 'claire.smith@scarlettbikes.com', '456-789-0123', 113);

For ora\_customers:

INSERT INTO ora\_customers (customer\_id, first\_name, last\_name, phone, email, street, city, state, zip\_code)

VALUES (1001, 'John', 'Doe', '123-456-7890', 'john.doe@example.com', '123 Main St', 'Springfield', 'IL', '62701');

INSERT INTO ora\_customers (customer\_id, first\_name, last\_name, phone, email, street, city, state, zip\_code)

VALUES (1002, 'Jane', 'Smith', '987-654-3210', 'jane.smith@example.com', '456 Elm St', 'Chicago', 'IL', '60616');

INSERT INTO ora\_customers (customer\_id, first\_name, last\_name, phone, email, street, city, state, zip\_code)

VALUES (1003, 'Mike', 'Brown', '555-123-4567', 'mike.brown@example.com', '789 Oak St', 'Houston', 'TX', '77001');

INSERT INTO ora\_customers (customer\_id, first\_name, last\_name, phone, email, street, city, state, zip\_code)

VALUES (1004, 'Sara', 'Johnson', '555-987-6543', 'sara.johnson@example.com', '321 Pine St', 'Seattle', 'WA', '98101');

For ora\_orders:

INSERT INTO ora\_orders (order\_id, customer\_id, order\_status, order\_date, required\_date, shipped\_date, store\_id, staff\_id)

VALUES (3001, 1001, 1, TO\_DATE('2024-12-05', 'YYYY-MM-DD'), TO\_DATE('2024-12-10', 'YYYY-MM-DD'), NULL, 111, 2001); -- Pending

INSERT INTO ora\_orders (order\_id, customer\_id, order\_status, order\_date, required\_date, shipped\_date, store\_id, staff\_id)

VALUES (3002, 1002, 2, TO\_DATE('2024-12-01', 'YYYY-MM-DD'), TO\_DATE('2024-12-06', 'YYYY-MM-DD'), TO\_DATE('2024-12-05', 'YYYY-MM-DD'), 112, 2002); -- Processing

INSERT INTO ora\_orders (order\_id, customer\_id, order\_status, order\_date, required\_date, shipped\_date, store\_id, staff\_id)

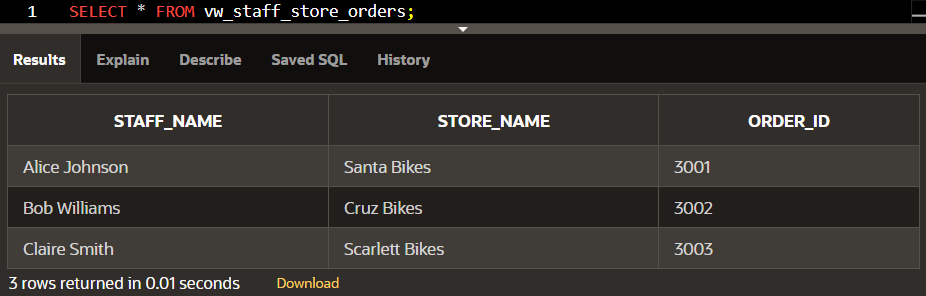
VALUES (3003, 1003, 3, TO\_DATE('2024-12-02', 'YYYY-MM-DD'), TO\_DATE('2024-12-08', 'YYYY-MM-DD'), NULL, 113, 2003); -- Rejected

INSERT INTO ora\_orders (order\_id, customer\_id, order\_status, order\_date, required\_date, shipped\_date, store\_id, staff\_id)

VALUES (3004, 1004, 4, TO\_DATE('2024-12-03', 'YYYY-MM-DD'), TO\_DATE('2024-12-09', 'YYYY-MM-DD'), TO\_DATE('2024-12-08', 'YYYY-MM-DD'), 113, 2003); -- Completed

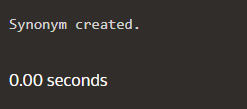
And more data but it will be too long to include.

Proof:



1. Create a synonym for the View created in (3e). We used:

CREATE SYNONYM syn\_staff\_store\_orders FOR vw\_staff\_store\_orders;



Proof:

