DBMS Final: Technical Report

BUAN 6320.503

Group 6

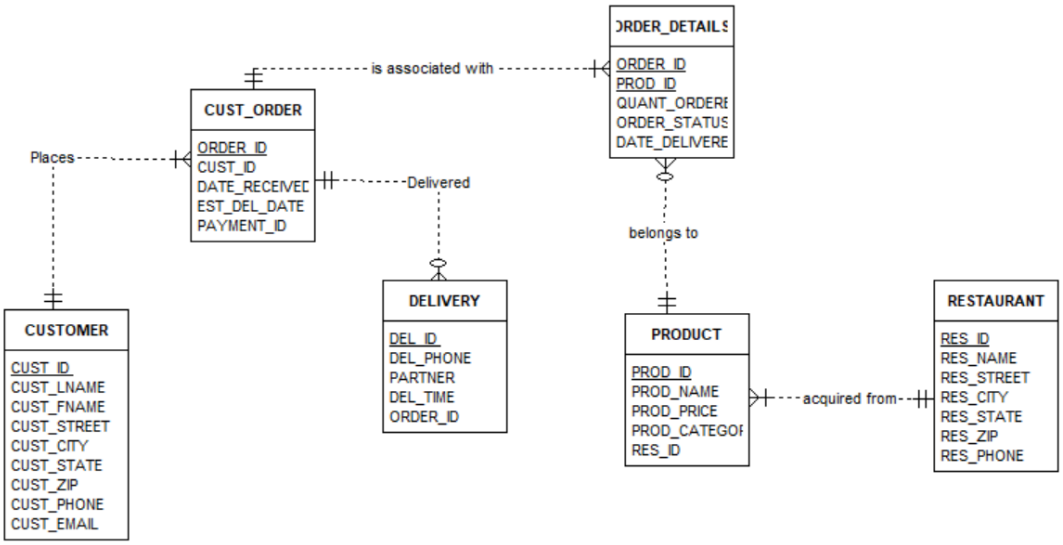
5 May 2022

Ujjval Bhatt,
Bharath Teja Bommidi,
Srikar Brahmadevara,
Christian Candler,
Twisha Choudhary,
Hemendra Gummadi,
Joyal Joby Chully,
Rahul Paladugula

Introduction

This Database Design Document describes the design and implementation of a database that will store user data for the online food orders with multiple restaurants. This is a centralized database in which several restaurants in the same location can be included and all orders can be tracked.

Entity Relationship Diagram:



Entity & Attribute Description:

Entity Name: **CUSTOMER**
Entity Description: A consumer of the restaurant's products.
Main Attributes of **CUSTOMER**:

- CUST_ID: A unique identifier for the customer. (Primary Key)
- CUST_LNAME: The last name of the customer.
- CUST_FNAME: The first name of the customer.
- CUST_STREET: The street from the customer's full address.
- CUST_CITY: The city from the customer's full address.
- CUST_STATE: The state from the customer's full address.
- CUST_ZIP: The zip code from the customer's full address.
- CUST_PHONE: The customer's phone number.
- CUST_EMAIL: The customer's email address.

Entity Name: **CUST_ORDER**
Entity Description: The order(s) that the customer places.
Main Attributes of **CUST_ORDER**:

- ORDER_ID: A unique identifier for the order placed. (Primary Key)
- CUST_ID: A unique identifier for the customer. (Foreign Key)
- DATE_RECEIVED: The date that the order is received.
- ES_DEL_DATE: The estimated order delivery date.
- PAYMENT_ID: The unique identifier of the order's payment.

Entity Name: **DELIVERY**
Entity Description: The delivery information for a customer's order.
Main Attributes of **DELIVERY**:

- DEL_ID: A unique identifier for the delivery. (Primary Key)
- DEL_PHONE: The phone number provided for the delivery.
- PARTNER: The service that the delivery will go through.
- DEL_TIME: The time of delivery.
- ORDER_ID: A unique identifier for the order. (Foreign Key)

Entity Name: **ORDER_DETAILS**
Entity Description: Details describing the order information in depth.
Main attributes of **ORDER_DETAILS**:

- ORDER_ID: A unique identifier for the order. (Primary Key)
- PROD_ID: A unique identifier for the product. (Foreign Key)
- QUANT_ORDERED: The exact quantity of items in the order.
- ORDER_STATUS: The status of the order.
- DATE_DELIVERED: The date that the order is confirmed to be delivered.

Entity Name: **PRODUCT**
Entity Description: The product that is being delivered to the customer.
Main Attributes of **PRODUCT:**

- PROD_ID: A unique identifier for the product. (Primary Key)
- PROD_NAME: The name of the product.
- PROD_PRICE: The price of a given product.
- PROD_CATEGORY: The category of item that the product falls under.
- RES_ID: A unique identifier for the restaurant. (Foreign Key)

Entity Name: **RESTAURANT**
Entity Description: The descriptors of the restaurant that the orders and products come from.
Main Attributes of **RESTAURANT**:

- RES_ID: A unique identifier for the restaurant. (Primary Key)
- RES_NAME: The name of the restaurant.
- RES_STREET: The street from the restaurant's full address.
- RES_CITY: The city from the restaurant's full address.
- RES_STATE: The state from the restaurant's full address.
- RES_ZIP: The zip code from the restaurant's full address.
- RES_PHONE: The phone number of the restaurant.

Relationship & Cardinality Description

Relationship: Customer places order
Cardinality: 1:M from Customer to Order
Business Rules:
A Customer may place one or many Customer Orders; A Customer Order must be placed by one and only one Customer

Relationship: Customer order delivered by Delivery(partner).
Cardinality: 1:M from Customer Order to Delivery
Business Rule: A Customer Order may be delivered 0, one or many delivery partners; One delivery can be made by one and only one Customer Order

Relationship: Customer order is associated with Order details
Cardinality: 1:M from Customer Order to Order Details
Business Rule: A Customer Order must be associated with one or many Order Details; One Order details can only be associated with one and only one Customer Order.

Relationship: Order Details belong to Product.
Cardinality: M:1 from Order Details to Product
Business Rule: One Order Detail belongs to one and only one Product; One Product can have zero, one or many Order Details

Relationship: Products had by the Restaurant.
Cardinality: 1:M between Restaurant and Product.
Business Rule: One Product may be acquired from one and only one Restaurant; One Restaurant can produce one or many products

Assumptions & Special Considerations

- Any Product will belong to a single restaurant in our platform
- A Customer can have two delivery options call in for order pick up at store and direct delivery to the location
- A customer can do multiple orders from multiple restaurants at a time.
- A delivery partner can deliver only one order at a time.

DDL

```
/* CREATE CUSTOMER table */
CREATE TABLE CUSTOMER(
   CUST_ID                    NUMBER(8) PRIMARY KEY,
   CUST_FNAME                 VARCHAR2(32) NOT NULL,
   CUST_LNAME                 VARCHAR2(32) NOT NULL,
   CUST_STREET                VARCHAR2(64) NOT NULL,
   CUST_CITY                  VARCHAR2(32) NOT NULL,
   CUST_STATE                 CHAR(2) NOT NULL,
   CUST_ZIP                   NUMBER(5) NOT NULL,
   CUST_PHONE                 CHAR(14) NOT NULL,
   CUST_EMAIL                 VARCHAR2(64) NOT NULL
);


/* CREATE RESTAURANT table*/
CREATE TABLE RESTAURANT(
   RES_ID          VARCHAR2(25) PRIMARY KEY,
   RES_NAME        VARCHAR2(32) NOT NULL,
   RES_STREET      VARCHAR2(32) NOT NULL,
   RES_CITY        VARCHAR2(16) NOT NULL,
   RES_STATE       CHAR(2) NOT NULL,
   RES_ZIP         NUMBER(5,0) NOT NULL,
   RES_PHONE       CHAR(14) NOT NULL
);

/* CREATE PRODUCT table*/
CREATE TABLE PRODUCT(
   PROD_ID                NUMBER(16,0) PRIMARY KEY,
   PROD_NAME              VARCHAR2(32) NOT NULL,
   PROD_PRICE             DECIMAL(8, 2) NOT NULL CHECK(PROD_PRICE > 0),
   PROD_CATEGORY          VARCHAR(30),
  RES_ID                  VARCHAR2(25) REFERENCES
            RESTAURANT(RES_ID)
);
```

```
/* CREATE CUST_ORDER table*/
CREATE TABLE CUST_ORDER(
   ORDER_ID                NUMBER(8) PRIMARY KEY,
   CUST_ID                 NUMBER(8) REFERENCES
              CUSTOMER(CUST_ID),
   DATE_RECEIVED           DATE,
   EST_DEL_DATE               DATE,
   PAYMENT_ID                 VARCHAR2(32) NOT NULL
);


/* CREATE ORDER_DETAILS table*/
CREATE TABLE ORDER_DETAILS(
   ORDER_ID                NUMBER(8) REFERENCES CUST_ORDER(ORDER_ID),
   PROD_ID                 NUMBER(16) REFERENCES PRODUCT,
   QUANT_ORDERED           INT DEFAULT 1 NOT NULL CHECK(QUANT_ORDERED > 0),
   ORDER_STATUS_ID         VARCHAR2(32) NOT NULL,
   DATE_DELIVERED          DATE,
   PRIMARY KEY(ORDER_ID, PROD_ID)
);

              /* CREATE DELIVERY table*/
CREATE TABLE DELIVERY(
   DEL_ID                  NUMBER(8) PRIMARY KEY,
   DEL_PHONE               CHAR(14) NOT NULL,
   PARTNER                 VARCHAR2(30) NOT NULL,
   DEL_TIME                VARCHAR2(10) NOT NULL,
   ORDER_ID                NUMBER(8) REFERENCES
              CUST_ORDER(ORDER_ID)
);

/* CREATE REST_PRODUCT view*/
CREATE VIEW REST_PROD AS
SELECT R.RES_ID, R.RES_NAME, P.PROD_NAME
FROM RESTAURANT R
LEFT JOIN PRODUCT P ON R.RES_ID=P.RES_ID;
```

```
/* CREATE SEQUENCE FOR DELIVERY PK*/
CREATE SEQUENCE SEQ_DEL
INCREMENT BY 1
START WITH 1
NOMAXVALUE
MINVALUE 0
NOCACHE;


/*CREATE TRIGGER FOR DELIVERY TABLE*/
CREATE OR REPLACE TRIGGER TRG_DEL
BEFORE INSERT OR UPDATE ON DELIVERY
FOR EACH ROW
BEGIN
IF INSERTING THEN
IF :NEW.DEL_ID IS NULL THEN
:NEW.DEL_ID := SEQ_DEL.NEXTVAL;
END IF;
END IF;
END;
```

Populate all tables

```
--CUSTOMER
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (101,'Luke', 'Hector', '678 East Willow', 'Dallas', 'TX', 75201, '588-999-
3333','lh@wolfhound.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (102,'Max', 'Hunter', '300 Lake Side', 'Dallas', 'TX', 75261, '918-298-
1888','hunter345@peaceliving.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (103,'Ben', 'Cook', '579 Hill Park', 'Richardson', 'TX', 75080, '257-281-
9123','bencook@gmail.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (104,'Harvey', 'Specter', '988 Palm Villas', 'Dallas', 'TX', 75204, '945-217-
0140','harvey_234@utdallas.edu');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (105,'Mike', 'Tyson', '276 Gold St', 'Richardson', 'TX', 75080, '276-123-
8799','lebronjames@fightclub.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (106,'Miccah', 'Paulson', '298 Laughing Villa', 'Dallas', 'TX', 75212, '908-908-
5199','micpaul@utdallas.edu');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (107,'Ameer', 'Suhaib', '10 Liscious Ave', 'Richardson', 'TX', 75080, '578-218-
9870','amsuh@lisciousnet.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (108,'Alex', 'Wilson', '21 Jump St', 'Richardson', 'TX', 75080, '999-999-
1234','alex.wil@utdallas.edu');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (109,'Ruth', 'Benjamin', '79 Forest Park', 'Dallas', 'TX', 75339, '123-455-
1234','ruth_benjamin@forestpark.com');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
VALUES (110,'Charles', 'Franklin', '980 Maple Dr', 'Richardson', 'TX', 75080, '734-437-
1792','charles789@utdallas.edu');
INSERT INTO CUSTOMER (CUST_ID, CUST_FNAME, CUST_LNAME, CUST_STREET,
CUST_CITY, CUST_STATE, CUST_ZIP, CUST_PHONE, CUST_EMAIL)
```

```
VALUES (111,'Mary', 'Heath', '714 Nettleship St', 'Dallas', 'TX', 75238, '214-438-
0890','MHeath1@aol.com');

--CUST_ORDER
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (1, 101, '2-05-2022', '5-05-2022' , 'AX56372877');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (2, 102, '2-05-2022', '5-05-2022',  'AX141633');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (3, 103, '3-05-2022', '5-05-2022',  'AX1824413');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (4, 104, '4-05-2022', '5-05-2022', 'AX633212');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (5, 105, '3-05-2022', '5-05-2022', 'AX633212');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (6, 106, '3-05-2022', '5-05-2022', 'AX418244212');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (7, 107, '3-05-2022', '5-05-2022', 'AX81522122');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (8, 108, '2-05-2022', '5-05-2022', 'AX9163311421');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (9, 109, '3-05-2022', '5-05-2022', 'AX91421615211');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (10, 110, '4-05-2022', '5-05-2022', 'AX1765218112');
INSERT INTO CUST_ORDER (ORDER_ID, CUST_ID, DATE_RECEIVED,
EST_DEL_DATE, PAYMENT_ID)
VALUES (11, 101, '4-05-2022', '5-05-2022', 'AX56372893');

--PRODUCT
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (21315, 'Taco', 6, 'FOOD', 1234);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (22118, 'Burrito', 12, 'FOOD', 1235);
```

```
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (51438, 'Enchilada', 11, 'FOOD', 1236);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (61109, 'Fajita', 8, 'FOOD', 1237);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (17215, 'Quesadilla', 9, 'FOOD', 1238);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (38112, 'Chalupa', 6, 'FOOD', 1239);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (20117, 'Taquito', 5, 'FOOD', 1240);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (19112, 'Salad', 7, 'FOOD', 1241);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (14138, 'Nachos', 4, 'FOOD', 1242);
INSERT INTO PRODUCT (PROD_ID, PROD_NAME, PROD_PRICE, PROD_CATEGORY,
RES_ID)
VALUES (38912, 'Chili', 6, 'FOOD', 1243);

--ORDER_DETAILS
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (1, 21315, 10, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (2, 22118, 12, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (3, 51438, 15, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (4, 61109, 20, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (5, 17215, 14, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (6, 38112, 19, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
```

```
VALUES (7, 20117, 20, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (8, 19112, 17, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (9, 14138, 14, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (10, 38912, 20, 'Delivered', '5-05-2022');
INSERT INTO ORDER_DETAILS (ORDER_ID, PROD_ID, QUANT_ORDERED,
ORDER_STATUS_ID, DATE_DELIVERED)
VALUES (11, 38912, 5, 'Delivered', '5-05-2022');

--RESTAURANT
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1234, 'Subway', 'Preston road', 'Richardson', 'TX', 75080, '999-999-9999');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1235, 'Wendys', 'Frankford road', 'Richardson', 'TX', 75250, '874-283-8426');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1236, 'Lezzet cafe', 'Frankford road', 'Richardson', 'TX', 75252, '469-931-2033');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1237, 'Taco bell', 'Coit road', 'Plano', 'TX', 75075, '469-626-0207');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1238, 'McDonalds', 'Coit road', 'Plano', 'TX', 75076, '972-758-0282');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1239, 'MashD', 'Preston road ste 1', 'Frisco', 'TX', 75034, '214-618-9440');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1240, 'Swadeshi plaza', 'Tx-121', 'Frisco', 'TX', 75035, '469-194-3500');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1241, 'Yard House', 'Las colinas', 'Irving', 'TX', 75039, '214-496-0151');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1242, 'Spice 6', 'Esters Rd', 'Irving', 'TX', 75061, '469-669-0329');
INSERT INTO RESTAURANT (RES_ID, RES_NAME, RES_STREET, RES_CITY,
RES_STATE, RES_ZIP, RES_PHONE)
VALUES (1243, 'Cadot Restaurant', 'Preston rd ste 120', 'Dallas', 'TX', 75252, '972-267-5700');
```

```
--DELIVERY
INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-862-9975', 'DOORDASH', '8:13PM', 1);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('972-802-5623 ', 'UBEREATS', '5:19PM', 2);

INSERT INTO DELIVERY (EL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-037-1963 ', 'UBEREATS', '7:03PM', 3);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-772-6252 ', 'POSTMATES', '6:24PM', 4);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('469-005-9078 ', 'DOORDASH', '7:49PM', 5);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-530-8520 ', 'DOORDASH', '9:19PM', 6);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('469-888-1115 ', 'POSTMATES', '5:34PM', 7);


INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('972-020-3300 ', 'UBEREATS', '5:57PM', 8);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-856-9007 ', 'DOORDASH', '7:16PM', 9);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('214-444-2307 ', 'POSTMATES', '6:33PM', 10);

INSERT INTO DELIVERY (DEL_PHONE, PARTNER, DEL_TIME, ORDER_ID)
VALUES ('469-484-2333 ', 'DOORDASH', '6:48PM', 11);
```

Basic Queries:

1. Select all columns and all rows from one table
   SELECT *
   FROM CUSTOMER;

```
SELECT * FROM CUSTOMER ORDER BY CUST_ID;
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	CUST_ZIP	CUST_PHONE	CUST_EMAIL
101	Luke	Hector	678 East Willow	Dallas	TX	75201	588-999-3333	lh@wolfhound.com
102	Max	Hunter	300 Lake Side	Dallas	TX	75261	918-298-1888	hunter345@peaceliving.com
103	Ben	Cook	579 Hill Park	Richardson	TX	75080	257-281-9123	bencook@gmail.com
104	Harvey	Specter	988 Palm Villas	Dallas	TX	75204	945-217-0140	harvey_234@utdallas.edu
105	Mike	Tyson	276 Gold St	Richardson	TX	75080	276-123-8799	lebronjames@fightclub.com
106	Miccah	Paulson	298 Laughing Villa	Dallas	TX	75212	908-908-5199	micpaul@utdallas.edu
107	Ameer	Suhaib	10 Liscious Ave	Richardson	TX	75080	578-218-9870	amsuh@lisciousnet.com
108	Alex	Wilson	21 Jump St	Richardson	TX	75080	999-999-1234	alex.wil@utdallas.edu
109	Ruth	Benjamin	79 Forest Park	Dallas	TX	75339	123-455-1234	ruth_benjamin@forestpark.com
110	Charles	Franklin	980 Maple Dr	Richardson	TX	75080	734-437-1792	charles789@utdallas.edu
111	Mary	Heath	714 Nettleship St	Dallas	TX	75238	214-438-0890	MHeath1@aol.com

2. Select five columns and all rows from one table
   SELECT CUST_ID, CUST_FNAME, CUST_CITY, CUST_PHONE, CUST_EMAIL
   FROM CUSTOMER;

```
SELECT CUST_ID, CUST_FNAME, CUST_CITY, CUST_PHONE, CUST_EMAIL
FROM CUSTOMER;
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_CITY	CUST_PHONE	CUST_EMAIL
101	Luke	Dallas	588-999-3333	lh@wolfhound.com
102	Max	Dallas	918-298-1888	hunter345@peaceliving.com
103	Ben	Richardson	257-281-9123	bencook@gmail.com
104	Harvey	Dallas	945-217-0140	harvey_234@utdallas.edu
105	Mike	Richardson	276-123-8799	lebronjames@fightclub.com
106	Miccah	Dallas	908-908-5199	micpaul@utdallas.edu
107	Ameer	Richardson	578-218-9870	amsuh@lisciousnet.com
108	Alex	Richardson	999-999-1234	alex.wil@utdallas.edu
109	Ruth	Dallas	123-455-1234	ruth_benjamin@forestpark.com
110	Charles	Richardson	734-437-1792	charles789@utdallas.edu
111	Mary	Dallas	214-438-0890	MHeath1@aol.com

3. Select all columns from all rows from one view
   SELECT * FROM REST_PROD

```
262  --3
263  SELECT * FROM REST_PROD
264
```

Results | Explain | Describe | Saved SQL | History

RES_ID	RES_NAME	PROD_NAME
1238	McDonalds	Quesadilla
1237	Taco bell	Fajita
1234	Subway	Taco
1235	Wendys	Burrito
1236	Lezzet cafe	Enchilada
1240	Swadeshi plaza	Taquito
1239	MashD	Chalupa
1241	Yard House	Salad
1242	Spice 6	Nachos
1243	Cadot Restaurant	Chili

10 rows returned in 0.02 seconds Download

4. Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product
   SELECT *
   FROM CUSTOMER
   LEFT OUTER JOIN CUST_ORDER
   ON CUSTOMER.CUST_ID = CUST_ORDER.CUST_ID
   ORDER BY CUSTOMER.CUST_ID;

```
1  SELECT *
2  FROM CUSTOMER
3  LEFT OUTER JOIN CUST_ORDER
4  ON CUSTOMER.CUST_ID = CUST_ORDER.CUST_ID
5  ORDER BY CUSTOMER.CUST_ID;
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	CUST_ZIP	CUST_PHONE	CUST_EMAIL	ORDER_ID	CUST_ID	DATE_RECEIVED	EST_DEL_DATE	PAYMENT_ID
101	Luke	Hector	678 East Willow	Dallas	TX	75201	588-999-3333	lh@wolfhound.com	1	101	02/05/0022	05/05/0022	AX56372877
101	Luke	Hector	678 East Willow	Dallas	TX	75201	588-999-3333	lh@wolfhound.com	11	101	04/05/0022	05/05/0022	AX56372893
102	Max	Hunter	300 Lake Side	Dallas	TX	75261	918-298-1888	hunter345@peaceliving.com	2	102	02/05/0022	05/05/0022	AX141633
103	Ben	Cook	579 Hill Park	Richardson	TX	75080	257-281-9123	bencook@gmail.com	3	103	03/05/0022	05/05/0022	AX1824413
104	Harvey	Specter	988 Palm Villas	Dallas	TX	75204	945-217-0140	harvey_234@utdallas.edu	4	104	04/05/0022	05/05/0022	AX633212
105	Mike	Tyson	276 Gold St	Richardson	TX	75080	276-123-8799	lebronjames@fightclub.com	5	105	03/05/0022	05/05/0022	AX633212
106	Miccah	Paulson	298 Laughing Villa	Dallas	TX	75212	908-908-5199	micpaul@utdallas.edu	6	106	03/05/0022	05/05/0022	AX418244212
107	Ameer	Suhaib	10 Liscious Ave	Richardson	TX	75080	578-218-9870	amsuh@lisciousnet.com	7	107	03/05/0022	05/05/0022	AX81522122
108	Alex	Wilson	21 Jump St	Richardson	TX	75080	999-999-1234	alex.wil@utdallas.edu	8	108	02/05/0022	05/05/0022	AX9163311421

5. Select and order data retrieved from one table
SELECT * FROM PRODUCT
ORDER BY PROD_ID;

```
272  --5
273  SELECT * FROM PRODUCT ORDER BY PROD_ID;
274
```

Results | Explain | Describe | Saved SQL | History

PROD_ID	PROD_NAME	PROD_PRICE	PROD_CATEGORY	RES_ID
14138	Nachos	4	FOOD	1242
17215	Quesadilla	9	FOOD	1238
19112	Salad	7	FOOD	1241
20117	Taquito	5	FOOD	1240
21315	Taco	6	FOOD	1234
22118	Burrito	12	FOOD	1235
38112	Chalupa	6	FOOD	1239
38912	Chili	6	FOOD	1243
51438	Enchilada	11	FOOD	1236
61109	Fajita	8	FOOD	1237

10 rows returned in 0.01 seconds Download

6. Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows
SELECT C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME, O.ORDER_ID, D.ORDER_STATUS_ID
FROM CUSTOMER C JOIN CUST_ORDER O ON C.CUST_ID = O.CUST_ID JOIN ORDER_DETAILS D ON O.ORDER_ID = D.ORDER_ID WHERE ROWNUM <= 10;

```
276  --6
277  SELECT C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME, O.ORDER_ID, D.ORDER_STATUS_ID
278  FROM CUSTOMER C JOIN CUST_ORDER O ON C.CUST_ID = O.CUST_ID JOIN ORDER_DETAILS D ON O.ORDER_ID = D.ORDER_ID WHERE ROWNUM <= 10;
279
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	ORDER_ID	ORDER_STATUS_ID
102	Max	Hunter	2	Delivered
101	Luke	Hector	1	Delivered
103	Ben	Cook	3	Delivered
107	Ameer	Suhaib	7	Delivered
108	Alex	Wilson	8	Delivered
105	Mike	Tyson	5	Delivered
110	Charles	Franklin	10	Delivered
104	Harvey	Specter	4	Delivered
109	Ruth	Benjamin	9	Delivered
106	Miccah	Paulson	6	Delivered

10 rows returned in 0.03 seconds Download

7. Select distinct rows using joins on 3 tables
   SELECT DISTINCT *
   FROM CUSTOMER C
   JOIN CUST_ORDER O
   ON C.CUST_ID = O.CUST_ID

```
281  --7
282  SELECT DISTINCT * FROM CUSTOMER C
283  JOIN CUST_ORDER O
284  ON C.CUST_ID = O.CUST_ID
285  JOIN ORDER_DETAILS D ON O.ORDER_ID = D.ORDER_ID;
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	CUST_ZIP	CUST_PHONE	CUST_EMAIL	ORDER_ID	CUST_ID	DATE_RECEIVED	EST_DEL_DATE
102	Max	Hunter	300 Lake Side	Dallas	TX	75261	918-298-1888	hunter345@peaceliving.com	2	102	02/05/2022	05/05/2022
101	Luke	Hector	678 East Willow	Dallas	TX	75201	588-999-3333	lh@wolfhound.com	1	101	02/05/2022	05/05/2022
103	Ben	Cook	579 Hill Park	Richardson	TX	75080	257-281-9123	bencook@gmail.com	3	103	03/05/2022	05/05/2022
107	Ameer	Suhaib	10 Liscious Ave	Richardson	TX	75080	578-218-9870	amsuh@lisciousnet.com	7	107	03/05/2022	05/05/2022
108	Alex	Wilson	21 Jump St	Richardson	TX	75080	999-999-1234	alex.wil@utdallas.edu	8	108	02/05/2022	05/05/2022
105	Mike	Tyson	276 Gold St	Richardson	TX	75080	276-123-8799	lebronjames@fightclub.com	5	105	03/05/2022	05/05/2022
110	Charles	Franklin	980 Maple Dr	Richardson	TX	75080	734-437-1792	charles789@utdallas.edu	10	110	04/05/2022	05/05/2022
104	Harvey	Specter	988 Palm Villas	Dallas	TX	75204	945-217-0140	harvey_234@utdallas.edu	4	104	04/05/2022	05/05/2022
109	Ruth	Benjamin	79 Forest Park	Dallas	TX	75339	123-455-1234	ruth_benjamin@forestpark.com	9	109	03/05/2022	05/05/2022
106	Miccah	Paulson	298 Laughing Villa	Dallas	TX	75212	908-908-5199	micpaul@utdallas.edu	6	106	03/05/2022	05/05/2022

More than 10 rows available. Increase rows selector to view more rows.

8. Use GROUP BY and HAVING in a select statement using one or more tables
   SELECT COUNT(CUST_ID) AS CUST_COUNT, CUST_CITY
   FROM CUSTOMER
   GROUP BY CUST_CITY
   HAVING COUNT(CUST_ID) >= 2;

```
288  --8
289  select count(cust_id) as cust_id, cust_city from customer
290  group by cust_city
291  having count(cust_id) >= 2;
292
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_CITY
5	Richardson
6	Dallas

2 rows returned in 0.01 seconds Download

9. Use IN clause to select data from one or more tables
   SELECT *
   FROM CUSTOMER
   WHERE CUST_CITY IN ('RICHARDSON', 'PLANO', 'DALLAS');

```
293  --9
294  select * from customer
295  where cust_city in ('Richardson', 'Plano', 'Dallas');
296
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	CUST_ZIP	CUST_PHONE	CUST_EMAIL
110	Charles	Franklin	980 Maple Dr	Richardson	TX	75080	734-437-1792	charles789@utdallas.edu
101	Luke	Hector	678 East Willow	Dallas	TX	75201	588-999-3333	lh@wolfhound.com
102	Max	Hunter	300 Lake Side	Dallas	TX	75261	918-298-1888	hunter345@peaceliving.com
103	Ben	Cook	579 Hill Park	Richardson	TX	75080	257-281-9123	bencook@gmail.com
104	Harvey	Specter	988 Palm Villas	Dallas	TX	75204	945-217-0140	harvey_234@utdallas.edu
107	Ameer	Suhaib	10 Liscious Ave	Richardson	TX	75080	578-218-9870	amsuh@lisciousnet.com
109	Ruth	Benjamin	79 Forest Park	Dallas	TX	75339	123-455-1234	ruth_benjamin@forestpark.com
105	Mike	Tyson	276 Gold St	Richardson	TX	75080	276-123-8799	lebronjames@fightclub.com
106	Miccah	Paulson	298 Laughing Villa	Dallas	TX	75212	908-908-5199	micpaul@utdallas.edu
108	Alex	Wilson	21 Jump St	Richardson	TX	75080	999-999-1234	alex.wil@utdallas.edu

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.02 seconds Download

10. Select length of one column from one table (use LENGTH function)
    SELECT LENGTH (CUST_FNAME) AS LENGTH
    FROM CUSTOMER;

```
298  --10
299  SELECT LENGTH (CUST_FNAME) AS LENGTH
300  FROM CUSTOMER;
301
```

Results | Explain | Describe | Saved SQL | History

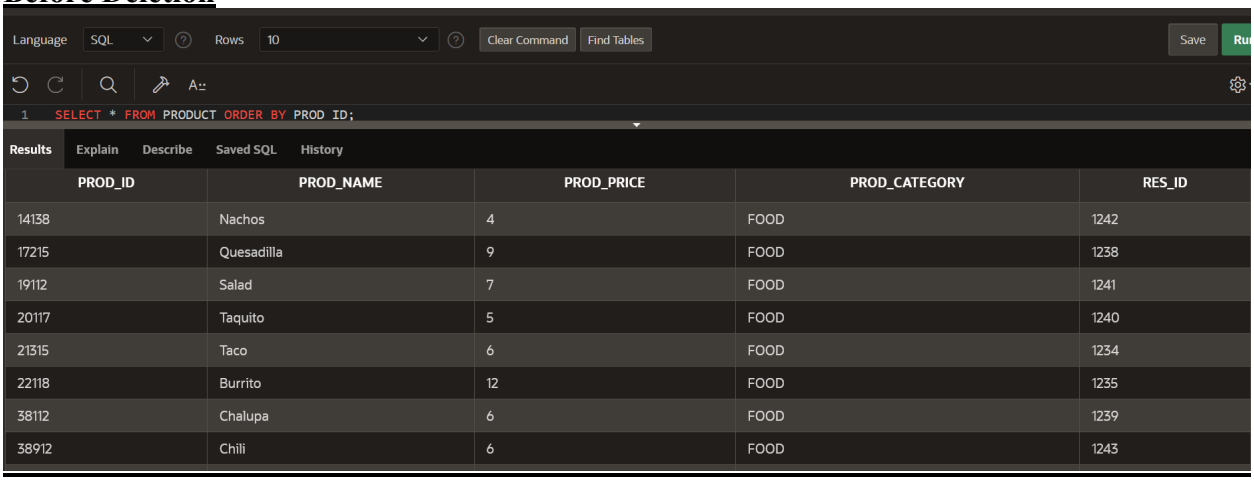
LENGTH
7
4
3
3
6
5
4
4
6
4

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.01 seconds Download

11. Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement.
SELECT * FROM PRODUCT ORDER BY PROD_ID;
DELETE FROM PRODUCT WHERE PROD_ID = 17215;

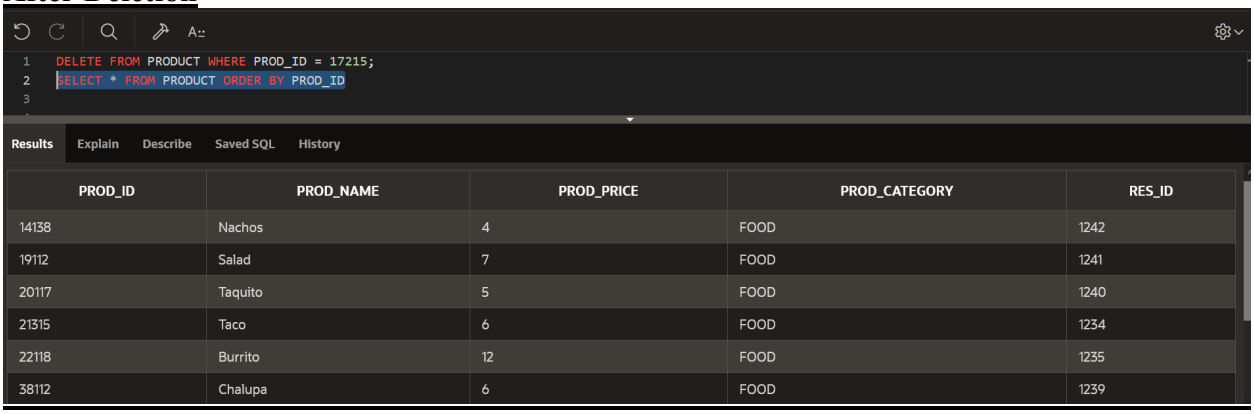
Before Deletion



PROD_ID	PROD_NAME	PROD_PRICE	PROD_CATEGORY	RES_ID
14138	Nachos	4	FOOD	1242
17215	Quesadilla	9	FOOD	1238
19112	Salad	7	FOOD	1241
20117	Taquito	5	FOOD	1240
21315	Taco	6	FOOD	1234
22118	Burrito	12	FOOD	1235
38112	Chalupa	6	FOOD	1239
38912	Chili	6	FOOD	1243

SELECT * FROM PRODUCT ORDER BY PROD_ID;

After Deletion



PROD_ID	PROD_NAME	PROD_PRICE	PROD_CATEGORY	RES_ID
14138	Nachos	4	FOOD	1242
19112	Salad	7	FOOD	1241
20117	Taquito	5	FOOD	1240
21315	Taco	6	FOOD	1234
22118	Burrito	12	FOOD	1235
38112	Chalupa	6	FOOD	1239

(Removed the row with Product id: 17215)

12. Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed
SELECT * FROM PRODUCT;
UPDATE PRODUCT SET PROD_NAME = 'Pasta' WHERE PROD_ID = 14138;

Before Updation

```
312  --12
313  SELECT * FROM PRODUCT;
314  UPDATE PRODUCT SET PROD_NAME = 'Pasta' WHERE PROD_ID = 14138;
315  SELECT * FROM PRODUCT;
316  ROLLBACK;
317
```

Results | Explain | Describe | Saved SQL | History

PROD_ID	PROD_NAME	PROD_PRICE	PROD_CATEGORY	RES_ID
17215	Quesadilla	9	FOOD	1238
61109	Fajita	8	FOOD	1237
21315	Taco	6	FOOD	1234
22118	Burrito	12	FOOD	1235
51438	Enchilada	11	FOOD	1236
20117	Taquito	5	FOOD	1240
38112	Chalupa	6	FOOD	1239
19112	Salad	7	FOOD	1241
14138	Nachos	4	FOOD	1242
38912	Chili	6	FOOD	1243

10 rows returned in 0.00 seconds Download

SELECT * FROM PRODUCT;
ROLLBACK;

After Updation

```
312  --12
313  SELECT * FROM PRODUCT;
314  UPDATE PRODUCT SET PROD_NAME = 'Pasta' WHERE PROD_ID = 14138;
315  SELECT * FROM PRODUCT;
316  ROLLBACK;
```

Results | Explain | Describe | Saved SQL | History

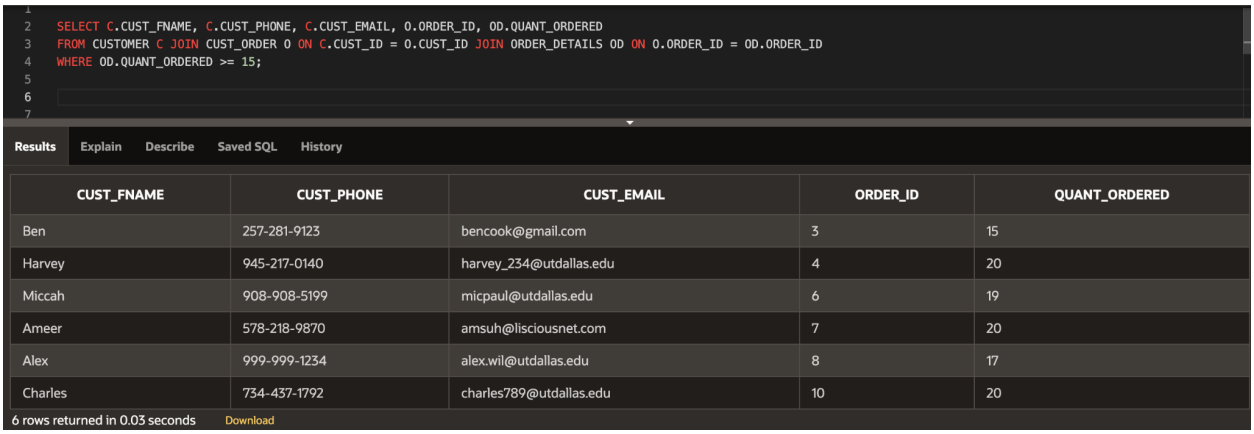
PROD_ID	PROD_NAME	PROD_PRICE	PROD_CATEGORY	RES_ID
17215	Quesadilla	9	FOOD	1238
61109	Fajita	8	FOOD	1237
21315	Taco	6	FOOD	1234
22118	Burrito	12	FOOD	1235
51438	Enchilada	11	FOOD	1236
20117	Taquito	5	FOOD	1240
38112	Chalupa	6	FOOD	1239
19112	Salad	7	FOOD	1241
14138	Pasta	4	FOOD	1242
38912	Chili	6	FOOD	1243

10 rows returned in 0.00 seconds Download

Advanced Queries:

1. Show customer details for orders with 15 or more items ordered

```
SELECT C.CUST_FNAME, C.CUST_PHONE, C.CUST_EMAIL,
O.ORDER_ID, OD.QUANT_ORDERED
FROM CUSTOMER C JOIN CUST_ORDER O ON C.CUST_ID =
O.CUST_ID JOIN ORDER_DETAILS OD ON O.ORDER_ID =
OD.ORDER_ID
WHERE OD.QUANT_ORDERED >= 15;
```



```
SELECT C.CUST_FNAME, C.CUST_PHONE, C.CUST_EMAIL, O.ORDER_ID, OD.QUANT_ORDERED
FROM CUSTOMER C JOIN CUST_ORDER O ON C.CUST_ID = O.CUST_ID JOIN ORDER_DETAILS OD ON O.ORDER_ID = OD.ORDER_ID
WHERE OD.QUANT_ORDERED >= 15;
```

Results | Explain | Describe | Saved SQL | History

CUST_FNAME	CUST_PHONE	CUST_EMAIL	ORDER_ID	QUANT_ORDERED
Ben	257-281-9123	bencook@gmail.com	3	15
Harvey	945-217-0140	harvey_234@utdallas.edu	4	20
Miccah	908-908-5199	micpaul@utdallas.edu	6	19
Ameer	578-218-9870	amsuh@lisciousnet.com	7	20
Alex	999-999-1234	alex.wil@utdallas.edu	8	17
Charles	734-437-1792	charles789@utdallas.edu	10	20

6 rows returned in 0.03 seconds Download

2. Show all deliveries to all customers

```
SELECT C.CUST_FNAME, C.CUST_LNAME, C.CUST_STREET,
      C.CUST_CITY, C.CUST_STATE,
      P.PROD_NAME, P.PROD_PRICE, OD.QUANT_ORDERED,
      P.PROD_PRICE * OD.QUANT_ORDERED TOTAL_PRICE,
      P.PROD_CATEGORY,
      D.DEL_TIME
FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID =
      P.RES_ID
    JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
    JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
    JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
    JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID;
```

```
SELECT C.CUST_FNAME, C.CUST_LNAME, C.CUST_STREET, C.CUST_CITY, C.CUST_STATE,
       P.PROD_NAME, P.PROD_PRICE, OD.QUANT_ORDERED, P.PROD_PRICE * OD.QUANT_ORDERED TOTAL_PRICE, P.PROD_CATEGORY,
       D.DEL_TIME
       FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID = P.RES_ID
        JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
        JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
        JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
        JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID;
```

Results | Explain | Describe | Saved SQL | History

CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	PROD_NAME	PROD_PRICE	QUANT_ORDERED	TOTAL_PRICE	PROD_CATEGORY	DEL_TIME
Charles	Franklin	980 Maple Dr	Richardson	TX	Chili	6	20	120	FOOD	6:33PM
Luke	Hector	678 East Willow	Dallas	TX	Taco	6	10	60	FOOD	8:13PM
Luke	Hector	678 East Willow	Dallas	TX	Chili	6	5	30	FOOD	6:48PM
Max	Hunter	300 Lake Side	Dallas	TX	Burrito	12	12	144	FOOD	5:19PM
Ben	Cook	579 Hill Park	Richardson	TX	Enchilada	11	15	165	FOOD	7:03PM
Harvey	Specter	988 Palm Villas	Dallas	TX	Fajita	8	20	160	FOOD	6:24PM
Ameer	Suhaib	10 Liscious Ave	Richardson	TX	Taquito	5	20	100	FOOD	5:34PM
Ruth	Benjamin	79 Forest Park	Dallas	TX	Nachos	4	14	56	FOOD	7:16PM
Mike	Tyson	276 Gold St	Richardson	TX	Quesadilla	9	14	126	FOOD	7:49PM
Miccah	Paulson	298 Laughing Villa	Dallas	TX	Chalupa	6	19	114	FOOD	9:19PM
Alex	Wilson	21 Jump St	Richardson	TX	Salad	7	17	119	FOOD	5:57PM

3. Show the list of customers from whom the total revenue generated is at least $100 ordered by revenue in descending order

```
SELECT O.CUST_ID, SUM(P.PROD_PRICE *
    OD.QUANT_ORDERED) AS REVENUE
FROM ORDER_DETAILS OD
JOIN PRODUCT P
ON OD.PROD_ID=P.PROD_ID
JOIN CUST_ORDER O
ON OD.ORDER_ID=O.ORDER_ID
GROUP BY O.CUST_ID
HAVING SUM(P.PROD_PRICE * OD.QUANT_ORDERED)>=100
ORDER BY REVENUE DESC;
```

```
--Query 15: Show the list of customers from whom the total revenue generated is at least $100 ordered by revenue in descending order

SELECT O.CUST_ID, SUM(P.PROD_PRICE * OD.QUANT_ORDERED) AS REVENUE
FROM ORDER_DETAILS OD
JOIN PRODUCT P
ON OD.PROD_ID=P.PROD_ID
JOIN CUST_ORDER O
ON OD.ORDER_ID=O.ORDER_ID
GROUP BY O.CUST_ID
HAVING SUM(P.PROD_PRICE * OD.QUANT_ORDERED)>=100
ORDER BY REVENUE DESC;
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	REVENUE
103	165
104	160
102	144
105	126
110	120
108	119
106	114
107	100

8 rows returned in 0.01 seconds Download

4. Show the customer's full name together and the area code of each customer's phone number.

```
SELECT CUST_FNAME || ' ' || CUST_LNAME AS CUST_NAME, '(' ||
SUBSTR(CUST_PHONE,1,3) || ')' AS AREA_CODE FROM
CUSTOMER;
```

```
--Query 16: Show the customer's full name together and the area code of each customer's phone number.

SELECT CUST_FNAME || ' ' || CUST_LNAME AS CUST_NAME, '(' || SUBSTR(CUST_PHONE,1,3) || ')' AS AREA_CODE FROM CUSTOMER;
```

Results | Explain | Describe | Saved SQL | History

CUST_NAME	AREA_CODE
Charles Franklin	(734)
Luke Hector	(588)
Max Hunter	(918)
Ben Cook	(257)
Harvey Specter	(945)
Ameer Suhaib	(578)
Ruth Benjamin	(123)
Mike Tyson	(276)
Miccah Paulson	(908)
Alex Wilson	(999)
Mary Heath	(214)

11 rows returned in 0.01 seconds Download

5. Show count of all deliveries based on city & delivery partner

```
SELECT C.CUST_CITY, C.CUST_STATE, D.PARTNER, COUNT(*)
    TOTAL_DELIVERIES
FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID =
    P.RES_ID
    JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
    JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
    JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
    JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID
GROUP BY C.CUST_CITY, C.CUST_STATE, D.PARTNER
ORDER BY COUNT(*) DESC;
```

```
--Query 17: Show count of all deliveries based on city & delivery partner

SELECT C.CUST_CITY, C.CUST_STATE, D.PARTNER, COUNT(*) TOTAL_DELIVERIES
FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID = P.RES_ID
    JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
    JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
    JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
    JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID
GROUP BY C.CUST_CITY, C.CUST_STATE, D.PARTNER
ORDER BY COUNT(*) DESC;
```

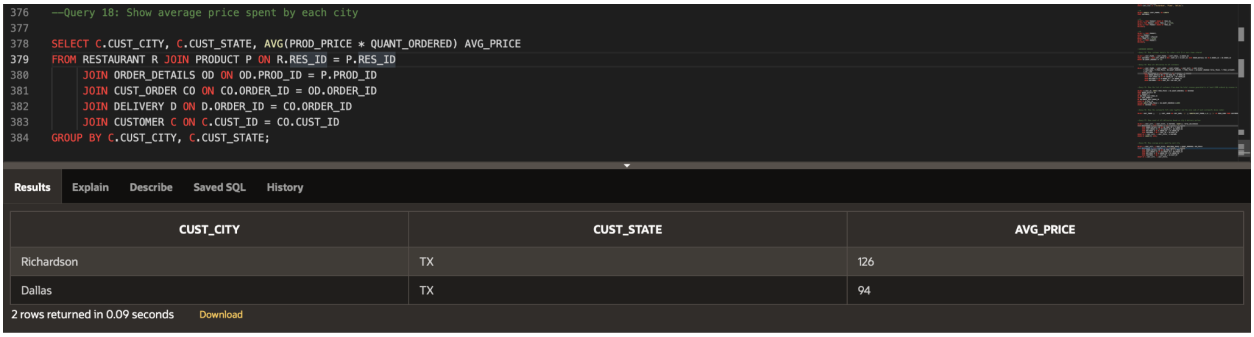
Results | Explain | Describe | Saved SQL | History

CUST_CITY	CUST_STATE	PARTNER	TOTAL_DELIVERIES
Dallas	TX	DOORDASH	4
Richardson	TX	POSTMATES	2
Richardson	TX	UBEREATS	2
Dallas	TX	UBEREATS	1
Richardson	TX	DOORDASH	1
Dallas	TX	POSTMATES	1

6 rows returned in 0.22 seconds Download

6. Show average price spent on an order by each city

```
SELECT C.CUST_CITY, C.CUST_STATE, AVG(PROD_PRICE *
      QUANT_ORDERED) AVG_PRICE
FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID =
      P.RES_ID
    JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
    JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
    JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
    JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID
GROUP BY C.CUST_CITY, C.CUST_STATE;
```



```
--Query 18: Show average price spent by each city

SELECT C.CUST_CITY, C.CUST_STATE, AVG(PROD_PRICE * QUANT_ORDERED) AVG_PRICE
FROM RESTAURANT R JOIN PRODUCT P ON R.RES_ID = P.RES_ID
      JOIN ORDER_DETAILS OD ON OD.PROD_ID = P.PROD_ID
      JOIN CUST_ORDER CO ON CO.ORDER_ID = OD.ORDER_ID
      JOIN DELIVERY D ON D.ORDER_ID = CO.ORDER_ID
      JOIN CUSTOMER C ON C.CUST_ID = CO.CUST_ID
GROUP BY C.CUST_CITY, C.CUST_STATE;
```

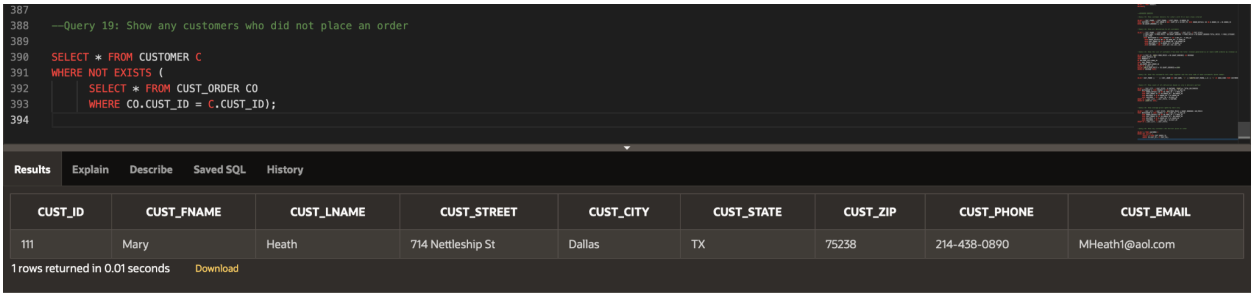
Results | Explain | Describe | Saved SQL | History

CUST_CITY	CUST_STATE	AVG_PRICE
Richardson	TX	126
Dallas	TX	94

2 rows returned in 0.09 seconds Download

7. Show any customers who did not place an order

```
SELECT *
FROM CUSTOMER C
WHERE NOT EXISTS (
    SELECT * FROM CUST_ORDER CO
    WHERE CO.CUST_ID = C.CUST_ID
);
```



```
--Query 19: Show any customers who did not place an order

SELECT * FROM CUSTOMER C
WHERE NOT EXISTS (
      SELECT * FROM CUST_ORDER CO
      WHERE CO.CUST_ID = C.CUST_ID);
```

Results | Explain | Describe | Saved SQL | History

CUST_ID	CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	CUST_ZIP	CUST_PHONE	CUST_EMAIL
111	Mary	Heath	714 Nettleship St	Dallas	TX	75238	214-438-0890	MHeath1@aol.com

1 rows returned in 0.01 seconds Download

8. List all customers who ordered and show total amount of orders each placed

```
SELECT C.CUST_FNAME, C.CUST_LNAME, C.CUST_STREET,
C.CUST_CITY, C.CUST_STATE, COUNT(*) TOTAL_ORDERS
FROM CUSTOMER C JOIN CUST_ORDER CO ON C.CUST_ID =
CO.CUST_ID
GROUP BY C.CUST_FNAME, C.CUST_LNAME,
C.CUST_STREET, C.CUST_CITY, C.CUST_STATE;
```

```
397  --Query 20: Show all customers who ordered and how many times
398
399  SELECT C.CUST_FNAME, C.CUST_LNAME, C.CUST_STREET, C.CUST_CITY, C.CUST_STATE, COUNT(*) TOTAL_ORDERS
400  FROM CUSTOMER C JOIN CUST_ORDER CO ON C.CUST_ID = CO.CUST_ID
401  GROUP BY C.CUST_FNAME, C.CUST_LNAME, C.CUST_STREET, C.CUST_CITY, C.CUST_STATE;
402
```

Results | Explain | Describe | Saved SQL | History

CUST_FNAME	CUST_LNAME	CUST_STREET	CUST_CITY	CUST_STATE	TOTAL_ORDERS
Mike	Tyson	276 Gold St	Richardson	TX	1
Ben	Cook	579 Hill Park	Richardson	TX	1
Ruth	Benjamin	79 Forest Park	Dallas	TX	1
Ameer	Suhaib	10 Liscious Ave	Richardson	TX	1
Luke	Hector	678 East Willow	Dallas	TX	2
Harvey	Specter	988 Palm Villas	Dallas	TX	1
Miccah	Paulson	298 Laughing Villa	Dallas	TX	1
Alex	Wilson	21 Jump St	Richardson	TX	1
Max	Hunter	300 Lake Side	Dallas	TX	1
Charles	Franklin	980 Maple Dr	Richardson	TX	1

10 rows returned in 0.04 seconds Download