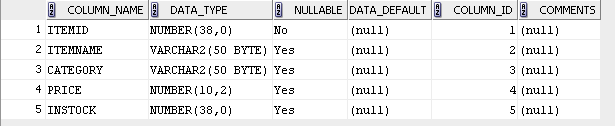
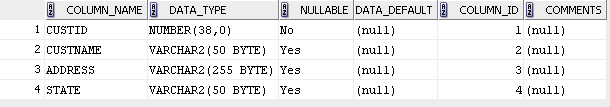
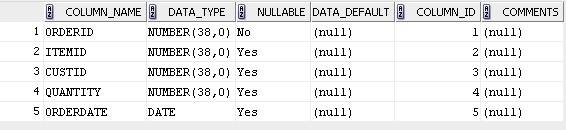
1)CREATE TABLE items\_exp9\_csa\_42( itemid INT PRIMARY KEY, itemname VARCHAR(50), category VARCHAR(50), price DECIMAL(10, 2), instock INT CHECK(instock >= 0));



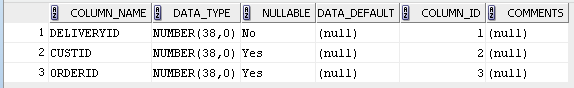
2CREATE TABLE customers\_exp9\_csa\_42(custid INT PRIMARY KEY,custname VARCHAR(50),address VARCHAR(255),state VARCHAR(50));



3)CREATE TABLE orders\_exp9\_csa\_42( orderid INT PRIMARY KEY, itemid INT, custid INT, quantity INT CHECK(quantity > 0), orderdate DATE, FOREIGN KEY (itemid) REFERENCES items\_exp9\_csa\_42(itemid), FOREIGN KEY (custid) REFERENCES customers\_exp9\_csa\_42(custid));

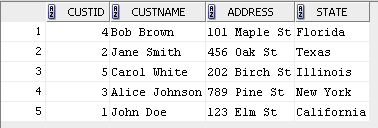


CREATE TABLE delivery\_exp9\_csa\_42( deliveryid INT PRIMARY KEY, custid INT, orderid INT, FOREIGN KEY (custid) REFERENCES customers\_exp9\_csa\_42(custid), FOREIGN KEY (orderid) REFERENCES orders\_exp9\_csa\_42(orderid));



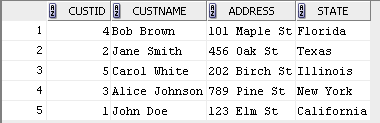
-- 1. List the details of all customers who have placed an order

SELECT DISTINCT c.\*FROM customers\_exp9\_csa\_42 cJOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid;



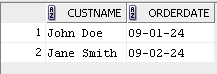
-- 2. List the details of all customers whose orders have been delivered

SELECT DISTINCT c.\*FROM customers\_exp9\_csa\_42 c JOIN delivery\_exp9\_csa\_42 d ON c.custid = d.custid;



-- 3. Find the order date for all customers whose name starts with the letter ‘J’

SELECT c.custname, o.orderdate FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custidWHERE c.custname LIKE 'J%';



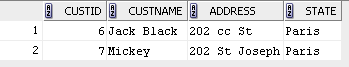
-- 4. Display the name and price of all items bought by the customer ‘Mickey’

SELECT i.itemname, i.priceFROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid JOIN items\_exp9\_csa\_42 i ON o.itemid = i.itemid WHERE c.custname = 'Mickey';



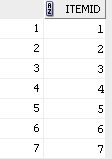
-- 5. List the details of all customers who have placed an order after January 2013 and not received delivery

SELECT DISTINCT c.\*FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid LEFT JOIN delivery\_exp9\_csa\_42 d ON o.orderid = d.orderid WHERE o.orderdate > '2013-01-01' AND d.orderid IS NULL;



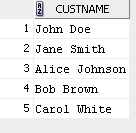
-- 6. Find the itemid of items which has either been ordered or not delivered (Use SET UNION)

SELECT itemid FROM orders\_exp9\_csa\_42 UNION SELECT i.itemid FROM items\_exp9\_csa\_42 I LEFT JOIN delivery\_exp9\_csa\_42d ON i.itemid = d.orderid WHERE d.orderid IS NULL;



-- 7. Find the custname of all customers who have placed an order and have their orders delivered (Use SET INTERSECTION)

SELECT c.custname FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custidJOIN delivery\_exp9\_csa\_42d ON o.orderid = d.orderid;



-- 8. Find the custname of all customers who have placed an order but not having their orders delivered (Use SET MINUS)

SELECT c.custname FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid LEFT JOIN delivery\_exp9\_csa\_42 d ON o.orderid = d.orderid WHERE d.orderid IS NULL;



-- 9. Find the name of the customer who has placed the most number of orders

SELECT c.custname, COUNT(o.orderid) AS total\_orders FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid GROUP BY c.custname ORDER BY total\_orders DESC LIMIT 1;

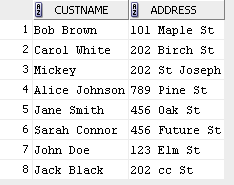


-- 10. Find the details of all customers who have purchased items exceeding a price of $5000

SELECT DISTINCT c.\* FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid JOIN items\_exp9\_csa\_42 i ON o.itemid = i.itemid WHERE i.price > 5000;

-- 11. Find the name and address of customers who have not ordered a ‘Samsung Galaxy S4’

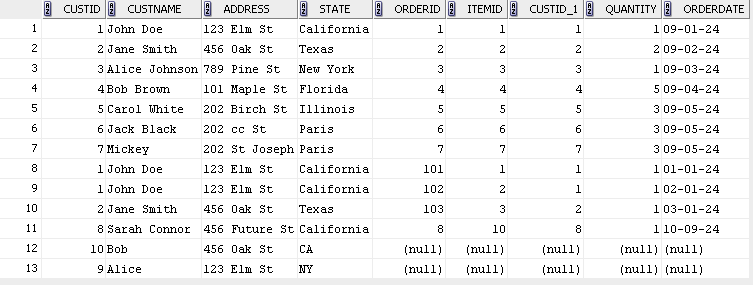
SELECT DISTINCT c.custname, c.address FROM customers\_exp9\_csa\_42 c JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid JOIN items\_exp9\_csa\_42 i ON o.itemid = i.itemid WHERE i.itemname != 'Samsung Galaxy S4';



-- 12. Perform Left Outer Join and Right Outer Join on Customers & Orders Table

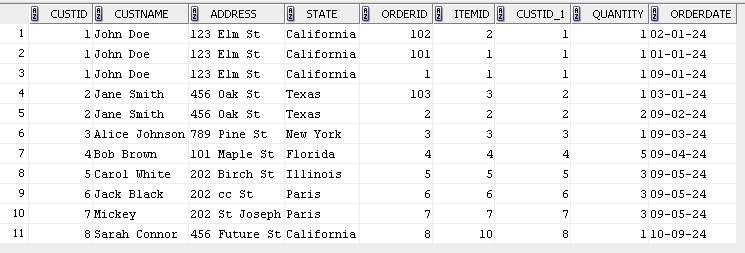
-- Left Outer Join (All customers and their orders, if any)

SELECT c.\*, o.\* FROM customers\_exp9\_csa\_42 c LEFT JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid;



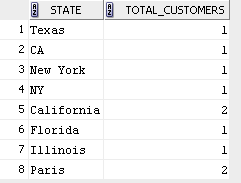
-- Right Outer Join (All orders and their corresponding customers, if any)

SELECT c.\*, o.\* FROM customers\_exp9\_csa\_42 c RIGHT JOIN orders\_exp9\_csa\_42 o ON c.custid = o.custid;



-- 13. Find the details of all customers grouped by state

SELECT state, COUNT(\*) AS total\_customers FROM customers\_exp9\_csa\_42 GROUP BY state;



-- 14. Display the details of all items grouped by category and having a price greater than the average price of all items

SELECT category, itemname, price FROM items\_exp9\_csa\_42 WHERE price > (SELECT AVG(price) FROM items\_exp9\_csa\_42) GROUP BY category, itemname, price;

