-- Create a table named orders

create table orders(

order\_no int primary key, purchase\_amount float, order\_date date,

customer\_id int, salesman\_id int);

-- Add values to the table

INSERT ALL

INTO orders VALUES(70001, 150.5, TO\_DATE('2012/10/05', 'YYYY/MM/DD'), 3005, 5002)

INTO orders VALUES(70009, 270.65, TO\_DATE('2012/09/10', 'YYYY/MM/DD'), 3001, 5005)

INTO orders VALUES(70002, 65.26, TO\_DATE('2012/10/05', 'YYYY/MM/DD'), 3002, 5001)

INTO orders VALUES(70004, 110.5, TO\_DATE('2012/08/17', 'YYYY/MM/DD'), 3009, 5003)

INTO orders VALUES(70007, 948.5, TO\_DATE('2012/09/10', 'YYYY/MM/DD'), 3005, 5002)

INTO orders VALUES(70005, 2400.6, TO\_DATE('2012/07/27', 'YYYY/MM/DD'), 3007, 5001)

INTO orders VALUES(70008, 5760, TO\_DATE('2012/08/15', 'YYYY/MM/DD'), 3002, 5001)

INTO orders VALUES(70010, 1983.43, TO\_DATE('2012/10/10', 'YYYY/MM/DD'), 3004, 5006)

INTO orders VALUES(70003, 2480.4, TO\_DATE('2012/10/10', 'YYYY/MM/DD'), 3009, 5003)

INTO orders VALUES(70012, 250.45, TO\_DATE('2012/06/27', 'YYYY/MM/DD'), 3008, 5002)

INTO orders VALUES(70011, 75.29, TO\_DATE('2012/08/17', 'YYYY/MM/DD'), 3003, 5007)

INTO orders VALUES(70013, 3045.6, TO\_DATE('2012/04/25', 'YYYY/MM/DD'), 3002, 5001)

SELECT 1 FROM DUAL;

-- Get all salesman ids without any repeated values

select distinct salesman\_id from orders;

-- Display the order number ordered by date in ascending order

select order\_no, order\_date from orders order by order\_date;

-- Display the order number ordered by purchase amount in descending order

select order\_no, purchase\_amount from orders order by purchase\_amount DESC;

-- Display the full data of orders that have purchase amount less than 500.

select \* from orders where purchase\_amount < 500;

-- Display the full data of orders that have purchase amount between 1000 and 2000.

select \* from orders where purchase\_amount between 1000 and 2000;

---------------Activity 7----------------------

--Write an SQL statement to find the total purchase amount of all orders.

select SUM(purchase\_amount) as sum from orders

--Write an SQL statement to find the average purchase amount of all orders.

select AVG(purchase\_amount) as avg from orders

--Write an SQL statement to get the maximum purchase amount of all the orders.

select MAX(purchase\_amount) as max from orders

--Write an SQL statement to get the minimum purchase amount of all the orders.

select MIN(purchase\_amount) as min from orders

--Write an SQL statement to find the number of salesmen listed in the table.

select COUNT(distinct salesman\_id) as count from orders

-----------------Activity 8-----------------------------

--Write an SQL statement to find the highest purchase amount ordered by the each customer with their ID and highest purchase amount.

select customer\_\_ID,max(purchase\_amount) as max from order group by customer\_id order by customer\_id

--Write an SQL statement to find the highest purchase amount on '2012-08-17' for each salesman with their ID.

select salesman\_id,order\_date,max(purchase\_amount) as max from orders

where order\_date = TO\_DATE('2012/08/17','YYYY/MM/DD') GROuP BY salesman\_id,order\_date;

--Write an SQL statement to find the highest purchase amount with their ID and order date, for only those customers who have a higher purchase amount within the list [2030, 3450, 5760, 6000].

select customer\_id,MAX(purchase\_amount) as max from orders Group by customer\_id having MAX(purchase\_amount) in (2030, 3450, 5760, 6000);

---------------Activity 6-11---------------------------

CREATE TABLE salesman (

salesman\_id int PRIMARY KEY,

salesman\_name varchar2(32),

salesman\_city varchar2(32),

commission int

);

--Activity 2

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

--VALUES (5001,'James Hoog','New York',15);

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

VALUES(5002,'Nail Knite','Paris',13)

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

VALUES(5005,'Pit Alex','London',11)

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

VALUES(5006,'McLyon','Paris',14)

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

VALUES (5007,'Paul Adam','Rome',13)

INSERT INTO salesman(salesman\_id,salesman\_name,salesman\_city,commission)

VALUES (5003,'Lauson Hen','San Jose',12);

INSERT All

INTO salesman VALUES(5001,'James Hoog','New York',15)

INTO salesman VALUES(5002,'Nail Knite','Paris',13)

INTO salesman VALUES(5003,'Lauson Hen','San Jose',12)

INTO salesman VALUES(5005,'Pit Alex','London',11)

INTO salesman VALUES(5006,'McLyon','Paris',14)

INTO salesman VALUES(5007,'Paul Adam','Rome',13)

SELECT 1 FROM DUAL;

select \* from orders;

-- Create the customers table

create table customers (

customer\_id int primary key, customer\_name varchar(32),

city varchar(20), grade int, salesman\_id int);

-- Insert values into it

INSERT ALL

INTO customers VALUES (3002, 'Nick Rimando', 'New York', 100, 5001)

INTO customers VALUES (3007, 'Brad Davis', 'New York', 200, 5001)

INTO customers VALUES (3005, 'Graham Zusi', 'California', 200, 5002)

INTO customers VALUES (3008, 'Julian Green', 'London', 300, 5002)

INTO customers VALUES (3004, 'Fabian Johnson', 'Paris', 300, 5006)

INTO customers VALUES (3009, 'Geoff Cameron', 'Berlin', 100, 5003)

INTO customers VALUES (3003, 'Jozy Altidor', 'Moscow', 200, 5007)

INTO customers VALUES (3001, 'Brad Guzan', 'London', 300, 5005)

SELECT 1 FROM DUAL;

select \* from salesman;

select \* from customers;

select \* from orders;

-- Create the customers table

create table customers (

customer\_id int primary key, customer\_name varchar(32),

city varchar(20), grade int, salesman\_id int);

-- Insert values into it

INSERT ALL

INTO customers VALUES (3002, 'Nick Rimando', 'New York', 100, 5001)

INTO customers VALUES (3007, 'Brad Davis', 'New York', 200, 5001)

INTO customers VALUES (3005, 'Graham Zusi', 'California', 200, 5002)

INTO customers VALUES (3008, 'Julian Green', 'London', 300, 5002)

INTO customers VALUES (3004, 'Fabian Johnson', 'Paris', 300, 5006)

INTO customers VALUES (3009, 'Geoff Cameron', 'Berlin', 100, 5003)

INTO customers VALUES (3003, 'Jozy Altidor', 'Moscow', 200, 5007)

INTO customers VALUES (3001, 'Brad Guzan', 'London', 300, 5005)

SELECT 1 FROM DUAL;

-----------Activity9----------------

--Write an SQL statement to know which salesman are working for which customer.

select customer\_id,salesman\_id from salesman s inner join customers c on s.salesman\_id=c.salesman\_id;

--Write an SQL statement to make a list of customers in ascending order with a salesman that have a grade less than 300.

select c.customer\_name,s.salesman\_name,c.grade from customers c inner join salesman s on c.salesman\_id=s.salesman\_id where c.grade < 300 order by s.SALESMAN\_NAME asc

--Write an SQL statement to find the list of customers who appointed a salesman who gets a commission of more than 12%.

select c.customer\_name,s.salesman\_name,s.commission from customers c inner join salesman s

on c.salesman\_id=s.salesman\_id where s.commission > 12 --order by s.SALESMAN\_NAME asc

--Write an SQL statement to find the following details of an order - order number, order date, purchase amount of order,

-- which customer gives the order and which salesman works for that customer and commission rate they get for the order.

SELECT a.order\_no, a.order\_date, a.purchase\_amount, b.customer\_name AS "Customer Name", b.grade, c.salesman\_name AS "Salesman", c.commission FROM orders a

INNER JOIN customers b ON a.customer\_id=b.customer\_id

INNER JOIN salesman c ON a.salesman\_id=c.salesman\_id;

----------------Activity 10-------------------------

--Write a query to find all the orders issued against the salesman who may works for customer whose id is 3007.

select \* from orders where salesman\_id in

(select s.salesman\_id from customers c INNER JOIN salesman s ON c.salesman\_id=s.salesman\_id where c.customer\_id=3007)

--Write a query to find all orders attributed to a salesman in New York.

select o.order\_no,o.purchase\_amount,o.order\_date from orders o INNER JOIN salesman s ON o.salesman\_id=s.salesman\_id where s.salesman\_city = 'New York'

--Write a query to count the customers with grades above New York's average.

select count(distinct customer\_id) from customers where grade > (select avg(grade) from customers where city = 'New York')

--Write a query to extract the data from the orders table for those salesman who earned the maximum commission

select o.order\_no,o.purchase\_amount,o.order\_date from salesman s inner join orders o on o.salesman\_id=s.salesman\_id

where s.commission = MAX(commission)

---------------------------Activity 11-----------------------

--Write a query that produces the name and number of each salesman and each customer with more than one current order.

--Put the results in alphabetical order.

SELECT customer\_id, customer\_name FROM customers a

WHERE 1<(SELECT COUNT(\*) FROM orders b WHERE a.customer\_id = b.customer\_id)

UNION

SELECT salesman\_id, salesman\_name FROM salesman a

WHERE 1<(SELECT COUNT(\*) FROM orders b WHERE a.salesman\_id = b.salesman\_id)

ORDER BY customer\_name;

--Write a query to make a report of which salesman produce the largest and smallest orders on each date. Also add a column that shows "highest on" and "lowest on" values.

SELECT a.salesman\_id, a.salesman\_name, o.order\_no, 'highest on', o.order\_date, o.purchase\_amount FROM salesman a, orders o

WHERE a.salesman\_id=o.salesman\_id

AND o.purchase\_amount=(SELECT MAX(purchase\_amount) FROM orders c WHERE c.order\_date = o.order\_date)

UNION

SELECT a.salesman\_id, a.salesman\_name, o.order\_no, 'lowest on', o.order\_date, o.purchase\_amount FROM salesman a, orders o

WHERE a.salesman\_id=o.salesman\_id

AND o.purchase\_amount=(SELECT MIN(purchase\_amount) FROM orders c WHERE c.order\_date = o.order\_date)

ORDER BY order\_date;