TABLES

***Sample table*: salesman**

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

***Sample table*: orders**

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007

***Sample table*: customer**

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

***Sample table*: nobel_win**

YEAR	SUBJECT	WINNER	COUNTRY	CATEGORY
1970	Physics	Hannes Alfven	Sweden	Scientist
1970	Physics	Louis Neel	France	Scientist
1970	Chemistry	Luis Federico Leloir	France	Scientist
1970	Physiology	Ulf von Euler	Sweden	Scientist
1970	Literature	Aleksandr Solzhenitsyn	Russia	Linguist
1970	Economics	Paul Samuelson	USA	Economist
1970	Physiology	Julius Axelrod	USA	Scientist
1971	Physics	Dennis Gabor	Hungary	Scientist
1971	Chemistry	Gerhard Herzberg	Germany	Scientist
1971	Peace	Willy Brandt	Germany	Chancellor
1971	Literature	Pablo Neruda	Chile	Linguist
1971	Economics	Simon Kuznets	Russia	Economist
1978	Peace	Anwar al-Sadat	Egypt	President
1978	Peace	Menachem Begin	Israel	Prime Minister
1987	Chemistry	Donald J. Cram	USA	Scientist
1987	Chemistry	Jean-Marie Lehn	France	Scientist
1987	Physiology	Susumu Tonegawa	Japan	Scientist
1994	Economics	Reinhard Selten	Germany	Economist
1994	Peace	Yitzhak Rabin	Israel	Prime Minister
1987	Physics	Johannes Georg Bednorz	Germany	Scientist
1987	Literature	Joseph Brodsky	Russia	Linguist
1987	Economics	Robert Solow	USA	Economist
1994	Literature	Kenzaburo Oe	Japan	Linguist

***Sample table*: item_mast**

```
PRO_ID PRO_NAME            PRO_PRICE  PRO_COM
------- ------------------ ---------- -------
    101 Mother Board       3200.00       15
    102 Key Board           450.00       16
    103 ZIP drive           250.00       14
    104 Speaker             550.00       16
    105 Monitor            5000.00       11
    106 DVD drive           900.00       12
    107 CD drive            800.00       12
    108 Printer            2600.00       13
    109 Refill cartridge    350.00       13
    110 Mouse               250.00       12
```

***Sample table*: emp_details**

```
EMP_IDNO EMP_FNAME       EMP_LNAME    EMP_DEPT
--------- --------------- ----------- --------
   127323 Michale         Robbin       57
   526689 Carlos          Snares       63
   843795 Enric           Dosio        57
   328717 Jhon            Snares       63
   444527 Joseph          Dosni        47
   659831 Zanifer         Emily        47
   847674 Kuleswar        Sitaraman    57
   748681 Henrey          Gabriel      47
   555935 Alex            Manuel       57
   539569 George          Mardy        27
   733843 Mario           Saule        63
   631548 Alan            Snappy       27
   839139 Maria           Foster       57
```

***Sample table*: testtable**

```
col1
--------------------------
A001/DJ-402\44_/100/2015
A001_\DJ-402\44_/100/2015
A001_DJ-402-2014-2015
A002_DJ-401-2014-2015
A001/DJ_401
A001/DJ_402\44
A001/DJ_402\44\2015
A001/DJ-402%45\2015/200
A001/DJ_402\45\2015%100
A001/DJ_402%45\2015/300
A001/DJ-402\44
```

***Sample table*: emp_department**

```
DPT_CODE DPT_NAME    DPT_ALLOTMENT
-------- ----------- -------------
      57 IT               65000
      63 Finance          15000
      47 HR               240000
      27 RD               55000
      89 QC               75000
```

***Sample table*: company_mast**

```
COM_ID COM_NAME
------ -------------
    11 Samsung
    12 iBall
    13 Epsion
    14 Zebronics
    15 Asus
    16 Frontech
```

1. Write a SQL statement that displays all the information about all salespeople.
2. Write a SQL statement to display a string "This is SQL Exercise, Practice and Solution".
3. Write a SQL query to display three numbers in three columns.
4. Write a SQL query to display the sum of two numbers 10 and 15 from the RDBMS server.
5. Write an SQL query to display the result of an arithmetic expression.
6. Write a SQL statement to display specific columns such as names and commissions for all salespeople.
7. Write a query to display the columns in a specific order, such as order date, salesman ID, order number, and purchase amount for all orders.
8. Write a SQL query to identify the unique salespeople ID. Return salesman_id.
9. Write a SQL query to locate salespeople who live in the city of 'Paris'. Return salesperson's name, city.
10. Write a SQL query to find customers whose grade is 200. Return customer_id, cust_name, city, grade, salesman_id.
11. Write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord_no, ord_date, purch_amt.
12. Write a SQL query to find the Nobel Prize winner(s) for the year 1970. Return year, subject and winner.
13. Write a SQL query to find the Nobel Prize winner in ‘Literature’ for 1971. Return winner.
14. Write a SQL query to locate the Nobel Prize winner ‘Dennis Gabor'. Return year, subject.
15. Write a SQL query to find the Nobel Prize winners in the field of ‘Physics’ since 1950. Return winner.

16. Write a SQL query to find the Nobel Prize winners in ‘Chemistry’ between the years 1965 and 1975. Begin and end values are included. Return year, subject, winner, and country.
17. Write a SQL query to display all details of the Prime Ministerial winners after 1972 of Menachem Begin and Yitzhak Rabin.
18. Write a SQL query to retrieve the details of the winners whose first names match with the string ‘Louis’. Return year, subject, winner, country, and category.
19. Write a SQL query that combines the winners in Physics, 1970 and in Economics, 1971. Return year, subject, winner, country, and category.
20. Write a SQL query to find the Nobel Prize winners in 1970 excluding the subjects of Physiology and Economics. Return year, subject, winner, country, and category.
21. Write a SQL query to combine the winners in 'Physiology' before 1971 and winners in 'Peace' on or after 1974. Return year, subject, winner, country, and category.
22. Write a SQL query to find the details of the Nobel Prize winner 'Johannes Georg Bednorz'. Return year, subject, winner, country, and category.
23. Write a SQL query to find Nobel Prize winners for the subject that does not begin with the letter 'P'. Return year, subject, winner, country, and category. Order the result by year, descending and winner in ascending.
24. Write a SQL query to find the details of 1970 Nobel Prize winners. Order the results by subject, ascending except for 'Chemistry' and ‘Economics’ which will come at the end of the result set. Return year, subject, winner, country, and category.
25. Write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com.
26. Write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.
27. Write a SQL query to display the pro_name as 'Item Name' and pro_priceas 'Price in Rs.'
28. Write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price.
29. Write a SQL query to calculate average price of the items for each company. Return average price and company code.
30. Write a SQL query to find the cheapest item(s). Return pro_name and, pro_price.
31. Write a SQL query to find the unique last name of all employees. Return emp_lname.
32. Write a SQL query to find the details of employees whose last name is 'Snares'. Return emp_idno, emp_fname, emp_lname, and emp_dept.
33. Write a SQL query to retrieve the details of the employees who work in the department 57. Return emp_idno, emp_fname, emp_lname and emp_dept..

Using Boolean and Relational operators

1. Write a SQL query to locate the details of customers with grade values above 100. Return customer_id, cust_name, city, grade, and salesman_id.
2. Write a SQL query to find all the customers in ‘New York’ city who have a grade value above 100. Return customer_id, cust_name, city, grade, and salesman_id.
3. Write a SQL query to find customers who are from the city of New York or have a grade of over 100. Return customer_id, cust_name, city, grade, and salesman_id.
4. Write a SQL query to find customers who are either from the city 'New York' or who do not have a grade greater than 100. Return customer_id, cust_name, city, grade, and salesman_id.
5. Write a SQL query to identify customers who do not belong to the city of 'New York' or have a grade value that exceeds 100. Return customer_id, cust_name, city, grade, and salesman_id.
6. Write a SQL query to find details of all orders excluding those with ord_date equal to '2012-09-10' and salesman_id higher than 5005 or purch_amt greater than 1000.Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
7. Write a SQL query to find the details of those salespeople whose commissions range from 0.10 to0.12. Return salesman_id, name, city, and commission.
8. Write a SQL query to find details of all orders with a purchase amount less than 200 or exclude orders with an order date greater than or equal to '2012-02-10' and a customer ID less than 3009. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
9. Write a SQL query to find all orders that meet the following conditions. Exclude combinations of order date equal to '2012-08-17' or customer ID greater than 3005 and purchase amount less than 1000.
10. Write a SQL query that displays order number, purchase amount, and the achieved and unachieved percentage (%) for those orders that exceed 50% of the target value of 6000.
11. Write a SQL query to find the details of all employees whose last name is ‘Dosni’ or ‘Mardy’. Return emp_idno, emp_fname, emp_lname, and emp_dept.

12. Write a SQL query to find the employees who work at depart 47 or 63. Return emp_idno, emp_fname, emp_lname, and emp_dept.

Wildcard and Special operators

1. Write a SQL query to find the details of those salespeople who come from the 'Paris' City or 'Rome' City. Return salesman_id, name, city, commission.
2. Write a SQL query to find the details of the salespeople who come from either 'Paris' or 'Rome'. Return salesman_id, name, city, commission.
3. Write a SQL query to find the details of those salespeople who live in cities other than Paris and Rome. Return salesman_id, name, city, commission.
4. Write a SQL query to retrieve the details of all customers whose ID belongs to any of the values 3007, 3008 or 3009. Return customer_id, cust_name, city, grade, and salesman_id.
5. Write a SQL query to find salespeople who receive commissions between 0.12 and 0.14 (begin and end values are included). Return salesman_id, name, city, and commission.
6. Write a SQL query to select orders between 500 and 4000 (begin and end values are included). Exclude orders amount 948.50 and 1983.43. Return ord_no, purch_amt, ord_date, customer_id, and salesman_id.
7. Write a SQL query to retrieve the details of the salespeople whose names begin with any letter between 'A' and 'L' (not inclusive). Return salesman_id, name, city, commission.
8. Write a SQL query to find the details of all salespeople except those whose names begin with any letter between 'A' and 'L' (not inclusive). Return salesman_id, name, city, commission.
9. Write a SQL query to retrieve the details of the customers whose names begins with the letter 'B'. Return customer_id, cust_name, city, grade, salesman_id..
10. Write a SQL query to find the details of the customers whose names end with the letter 'n'. Return customer_id, cust_name, city, grade, salesman_id.
11. Write a SQL query to find the details of those salespeople whose names begin with ‘N’ and the fourth character is 'l'. Rests may be any character. Return salesman_id, name, city, commission.
12. Write a SQL query to find those rows where col1 contains the escape character underscore (_). Return col1.
13. Write a SQL query to identify those rows where col1 does not contain the escape character underscore (_). Return col1.
14. Write a SQL query to find rows in which col1 contains the forward slash character (/). Return col1.
15. Write a SQL query to identify those rows where col1 does not contain the forward slash character (/). Return col1.
16. Write a SQL query to find those rows where col1 contains the string (_/). Return col1.
17. Write a SQL query to find those rows where col1 does not contain the string (_/). Return col1.
18. From the following table, write a SQL query to find those rows where col1 contains the character percent (%). Return col1.
19. From the following table, write a SQL query to find those rows where col1 does not contain the character percent (%). Return col1.
20. Write a SQL query to find all those customers who does not have any grade. Return customer_id, cust_name, city, grade, salesman_id.
21. Write a SQL query to locate all customers with a grade value. Return customer_id, cust_name,city, grade, salesman_id.
22. Write a SQL query to locate the employees whose last name begins with the letter 'D'. Return emp_idno, emp_fname, emp_lname and emp_dept.

Functions and Group by

1. Write a SQL query to calculate total purchase amount of all orders. Return total purchase amount.

2. Write a SQL query to calculate the average purchase amount of all orders. Return average purchase amount.

3. Write a SQL query that counts the number of unique salespeople. Return number of salespeople.

4. Write a SQL query to count the number of customers. Return number of customers.

5. Write a SQL query to determine the number of customers who received at least one grade for their activity.

6. Write a SQL query to find the maximum purchase amount.

7. Write a SQL query to find the minimum purchase amount.

8. Write a SQL query to find the highest grade of the customers in each city. Return city, maximum grade.

9. Write a SQL query to find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount.

10. Write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.

11. Write a SQL query to determine the highest purchase amount made by each salesperson on '2012-08-17'. Return salesperson ID, purchase amount

12. Write a SQL query to find the highest order (purchase) amount by each customer on a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount.

13. Write a SQL query to find the maximum order (purchase) amount in the range 2000 - 6000 (Begin and end values are included.) By combination of each customer and order date. Return customer id, order date and maximum purchase amount.

14. Write a SQL query to find the maximum order (purchase) amount based on the combination of each customer and order date. Filter the rows for maximum order (purchase) amount is either 2000, 3000, 5760, 6000. Return customer id, order date and maximum purchase amount.

15. Write a SQL query to determine the maximum order amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount.

16. Write a SQL query to find the maximum order (purchase) amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Filter the rows for maximum order (purchase) amount is higher than 1000. Return customer id and maximum purchase amount.

17. Write a SQL query to determine the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount.

18. Write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders.

19. Write a SQL query to count the number of salespeople in a city. Return number of salespeople.

20. Write a SQL query to count the number of orders based on the combination of each order date and salesperson. Return order date, salesperson id.

21. Write a SQL query to calculate the average product price. Return average product price.

22. , write a SQL query to count the number of products whose price are higher than or equal to 350. Return number of products.

23. Write a SQL query to compute the average price for unique companies. Return average price and company id.

24. Write a SQL query to compute the sum of the allotment amount of all departments. Return sum of the allotment amount.

25. Write a SQL query to count the number of employees in each department. Return department code and number of employees.

Formatting Output

1. Write a SQL query to select all the salespeople. Return salesman_id, name, city, commission with the percent sign (%).

2. Write a SQL query to find the number of orders booked for each day. Return the result in a format like "For 2001-10-10 there are 15 orders".".

3. Write a SQL query to find all the orders. Sort the result-set in ascending order by ord_no. Return all fields.

4. Write a SQL query to find all the orders. Sort the result-set in descending order by ord_date. Return all fields.

5. Write a SQL query to find all the orders. Sort the result-set in descending order by ord_date and purch_amt. Return all fields.

6. Write a SQL query to find all the customers. Sort the result-set by customer_id. Return cust_name, city, grade.

7. Write a SQL query that calculates the maximum purchase amount generated by each salesperson for each order date. Sort the result-set by salesperson id and order date in ascending order. Return salesperson id, order date and maximum purchase amount.

8. Write a SQL query to find all the customers. Sort the result-set in descending order on 3rd field. Return customer name, city and grade.

9. Write a SQL query that counts the unique orders and the highest purchase amount for each customer. Sort the result-set in descending order on 2nd field. Return customer ID, number of distinct orders and highest purchase amount by each customer.

10. Write a SQL query to calculate the summation of purchase amount, total commission (15% for all salespeople) by each order date. Sort the result-set on order date. Return order date, summation of purchase amount and commission.

Query on Multiple Tables

1. Write a SQL query to find the salespeople and customers who live in the same city. Return customer name, salesperson name and salesperson city.

2. Write a SQL query to locate all the customers and the salesperson who works for them. Return customer name, and salesperson name.

3. Write a SQL query to find those salespeople who generated orders for their customers but are not located in the same city. Return ord_no, cust_name, customer_id (orders table), salesman_id (orders table).

4. Write a SQL query to locate the orders made by customers. Return order number, customer name.

5. Write a SQL query to find those customers where each customer has a grade and is served by a salesperson who belongs to a city. Return cust_name as "Customer", grade as "Grade".

6. Write a SQL query to find those customers who are served by a salesperson and the salesperson earns commission in the range of 12% to 14% (Begin and end values are included.). Return cust_name AS "Customer", city AS "City".

7. Write a SQL query to find all orders executed by the salesperson and ordered by the customer whose grade is greater than or equal to 200. Compute purch_amt*commission as “Commission”. Return customer name, commission as “Commission%” and Commission.

8. Write a SQL query to find those customers who placed orders on October 5, 2012. Return customer_id, cust_name, city, grade, salesman_id, ord_no, purch_amt, ord_date, customer_id and salesman_id.

JOINS

1. Write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city.

2. Write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

3. Write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission.

4. Write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

5. Write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission.

6. Write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission.

7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

8. Write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

9. Write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.

10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not.

11. SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves.

12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers.

13. Write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

15. For those customers from the existing list who put one or more orders, or which orders have been placed by the customer who is not on the list, create a report containing the customer name, city, order number, order date, and purchase amount

16. Write a SQL statement to generate a report with the customer name, city, order no. Order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

17. Write a SQL query to combine each row of the salesman table with each row of the customer table.

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. Each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city.

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. Each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade.

20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. Each salesman will appear for all customers and vice versa for those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade.

21. Write a SQL query to select all rows from both participating tables as long as there is a match between pro_com and com_id.

22. Write a SQL query to display the item name, price, and company name of all the products.

23. Write a SQL query to calculate the average price of items of each company. Return average value and company name.

24. Write a SQL query to calculate and find the average price of items of each company higher than or equal to Rs. 350. Return average value and company name.

25. Write a SQL query to find the most expensive product of each company. Return pro_name, pro_price and com_name.

26. Write a SQL query to display all the data of employees including their department.

27. Write a SQL query to display the first and last names of each employee, as well as the department name and sanction amount.

28. Write a SQL query to find the departments with budgets more than Rs. 50000 and display the first name and last name of employees.

29. Write a SQL query to find the names of departments where more than two employees are employed. Return dpt_name.

SUBQUERIES

1. Write a SQL query to find all the orders issued by the salesman 'Paul Adam'. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
2. Write a SQL query to find all orders generated by London-based salespeople. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.
3. Write a SQL query to find all orders generated by the salespeople who may work for customers whose id is 3007. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.
4. Write a SQL query to find the order values greater than the average order value of 10th October 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.
5. Write a SQL query to find all the orders generated in New York city. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
6. Write a SQL query to determine the commission of the salespeople in Paris. Return commission.
7. Write a query to display all the customers whose ID is 2001 below the salesperson ID of Mc Lyon.
8. Write a SQL query to count the number of customers with grades above the average in New York City. Return grade and count.

9. Write a SQL query to find those salespeople who earned the maximum commission. Return ord_no, purch_amt, ord_date, and salesman_id.
10. Write SQL query to find the customers who placed orders on 17th August 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id and cust_name.
11. Write a SQL query to find salespeople who had more than one customer. Return salesman_id and name.
12. Write a SQL query to find those orders, which are higher than the average amount of the orders. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
13. Write a SQL query to find those orders that are equal or higher than the average amount of the orders. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
14. Write a query to find the sums of the amounts from the orders table, grouped by date, and eliminate all dates where the sum was not at least 1000.00 above the maximum order amount for that date.
15. Write a query to extract all data from the customer table if and only if one or more of the customers in the customer table are located in London.
16. Write a SQL query to find salespeople who deal with multiple customers. Return salesman_id, name, city and commission.
17. Write a SQL query to find salespeople who deal with a single customer. Return salesman_id, name, city and commission.
18. Write a SQL query to find the salespeople who deal the customers with more than one order. Return salesman_id, name, city and commission.
19. Write a SQL query to find the salespeople who deal with those customers who live in the same city. Return salesman_id, name, city and commission.
20. Write a SQL query to find salespeople whose place of residence matches any city where customers live. Return salesman_id, name, city and commission.
21. Write a SQL query to find all those salespeople whose names appear alphabetically lower than the customer's name. Return salesman_id, name, city, commission.
22. Write a SQL query to find all those customers with a higher grade than all the customers alphabetically below the city of New York. Return customer_id, cust_name, city, grade, salesman_id.
23. Write a SQL query to find all those orders whose order amount exceeds at least one of the orders placed on September 10th 2012. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
24. Write a SQL query to find orders where the order amount is less than the order amount of a customer residing in London City. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
25. Write a SQL query to find those orders where every order amount is less than the maximum order amount of a customer who lives in London City. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.
26. Write a SQL query to find those customers whose grades are higher than those living in New York City. Return customer_id, cust_name, city, grade and salesman_id.
27. Write a SQL query to calculate the total order amount generated by a salesperson. Salespersons should be from the cities where the customers reside. Return salesperson name, city and total order amount.
28. Write a SQL query to find those customers whose grades are not the same as those who live in London City. Return customer_id, cust_name, city, grade and salesman_id.
29. Write a SQL query to find those customers whose grades are different from those living in Paris. Return customer_id, cust_name, city, grade and salesman_id.
30. Write a SQL query to find all those customers who have different grades than any customer who lives in Dallas City. Return customer_id, cust_name,city, grade and salesman_id.
31. Write a SQL query to calculate the average price of each manufacturer's product along with their name. Return Average Price and Company.
32. Write a SQL query to calculate the average price of each manufacturer's product of 350 or more. Return Average Price and Company.
33. Write a SQL query to find the most expensive product of each company. Return Product Name, Price and Company.
34. Write a SQL query to find employees whose last name is Gabriel or Dosio. Return emp_idno, emp_fname, emp_lname and emp_dept.
35. Write a SQL query to find the employees who work in department 89 or 63. Return emp_idno, emp_fname, emp_lname and emp_dept.
36. Write a SQL query to find those employees who work for the department where the departmental allotment amount is more than Rs. 50000. Return emp_fname and emp_lname.
37. Write a SQL query to find the departments whose sanction amount is higher than the average sanction amount for all departments. Return dpt_code, dpt_name and dpt_allotment.
38. Write a SQL query to find which departments have more than two employees. Return dpt_name.
39. Write a SQL query to find the departments with the second lowest sanction amount. Return emp_fname and emp_lname.

UNION

1. Write a SQL query to find all salespeople and customers located in the city of London.
2. Write a SQL query to find distinct salespeople and their cities. Return salesperson ID and city.
3. Write a SQL query to find all those salespeople and customers who are involved in the inventory management system. Return salesperson ID, customer ID.
4. Write a SQL query to find the salespersons who generated the largest and smallest orders on each date. Return salesperson ID, name, order no., highest on/lowest on, order date.
5. Write a SQL query to find the salespeople who generated the largest and smallest orders on each date. Sort the result-set on third field. Return salesperson ID, name, order no., highest on/lowest on, order date.
6. Write a SQL query to find those salespeople who live in the same city where the customer lives as well as those who do not have customers in their cities by indicating 'NO MATCH'. Sort the result set on 2nd column (i.e. Name) in descending order. Return salesperson ID, name, customer name, commission.
7. Write a SQL query that appends strings to the selected fields, indicating whether the city of any salesperson is matched with the city of any customer. Return salesperson ID, name, city, MATCHED/NO MATCH.
8. Write a SQL query to create a union of two queries that shows the customer id, cities, and ratings of all customers. Those with a rating of 300 or greater will have the words 'High Rating', while the others will have the words 'Low Rating'.
9. Write a SQL query to find those salespersons and customers who have placed more than one order. Return ID, name.

VIEW

1. Create a view for those salespeople who belong to the city of New York.
2. Create a view for all salespersons. Return salesperson ID, name, and city.
3. Create a view to locate the salespeople in the city 'New York'.
4. Create a view that counts the number of customers in each grade.
5. Create a view to count the number of unique customers, compute the average and the total purchase amount of customer orders by each date.
6. Create a view to get the salesperson and customer by name. Return order name, purchase amount, salesperson ID, name, customer name.
7. Create a view to find the salesperson who handles a customer who makes the highest order of the day. Return order date, salesperson ID, name.
8. Create a view to find the salesperson who deals with the customer with the highest order at least three times per day. Return salesperson ID and name.
9. Create a view to find all the customers who have the highest grade. Return all the fields of customer.
10. Create a view to count the number of salespeople in each city. Return city, number of salespersons.
11. Create a view to compute the average purchase amount and total purchase amount for each salesperson. Return name, average purchase and total purchase amount. (Assume all names are unique.).
12. Create a view to identify salespeople who work with multiple clients. Return all the fields of salesperson.
13. Create a view that shows all matching customers with salespeople, ensuring that at least one customer in the city of the customer is served by the salesperson in the city of the salesperson.
14. Create a view to display the number of orders per day. Return order date and number of orders.
15. Create a view to find the salespeople who placed orders on October 10th, 2012. Return all the fields of salesperson.
16. Create a view to find the salespersons who issued orders on either August 17th, 2012 or October 10th, 2012. Return salesperson ID, order number and customer ID.