**ASSIGNMENT 1**

* **CREATE 'STUDENT\_DB' DATABASE**

CREATE DATABASE STUDENT\_DB;

* **CREATE FOLLOWING TABLES IN DATABASE WITH PROPER DATA TYPES**
* **STUDENT: ID (PK), NAME, ADDRESS, PHONE, EMAIL, AGE, SCHOOLID (FK)**
* **SCHOOL: ID (PK), NAME, ADDRESS, PRINCIPAL, PHONE**

CREATE TABLE STUDENT(ID INT PRIMARY KEY IDENTITY(1,1),NAME VARCHAR(20),ADDRESS VARCHAR(40),PHONE VARCHAR(12),EMAIL VARCHAR(25),AGE INT,SCHOOLID INT ,FOREIGN KEY(SCHOOLID) REFERENCES SCHOOL(ID));

CREATE TABLE SCHOOL(ID INT PRIMARY KEY IDENTITY(1,1),NAME VARCHAR(20),ADDRESS VARCHAR(30),PRICIPAL VARCHAR(20),PHONE VARCHAR(12));

* **CREATE RELATIONSHIP AS SHOWN IN THE TABLE SCHEMA**

DONE

* **ADD SOME DUMMY DATA**

INSERT INTO STUDENT(NAME,ADDRESS,PHONE,EMAIL,AGE,SCHOOLID) VALUES('ABHAY PANCHAL','SHIVAJI CHOWK, NILANGA','9146334963','ABHAYRPANCHAL@GMAIL.COM',26,2),('SHUBHAM WAGHMARE','ANAND CHOWK, LATUR','9850800961','SHUBHAMWAGHMARE@GMAIL.COM',26,2),('SHUBHAM PANCHAL','SHIVAJI CHOWK, NILANGA','9767356855','SHUBHAMPANCHAL@GMAIL.COM',24,3),('PRANAV PANCHAL','ANANDMUNI CHOWK, NILANGA','9146334963','PRANAVPANCHAL@GMAIL.COM',20,2),('RAJ MISHRA','BUDHH NAGAR, NANDED','9167634683','RAKMISHRA@GMAIL.COM',24,1),('SURAJ KUMAR','HATTI CHOWK,PUNE','9147394528','SURAJ123@GMAIL.COM',25,3);

**ASSIGNMENT 2**

* **CREATE 'PRODUCT\_DB' DATABASE**

CREATE DATABASE PRODUCT\_DB;

USE PRODUCT\_DB;

* **CREATE FOLLOWING TABLES**
* **CATEGORIES: ID (PK), TITLE, DESCRIPTION**
* **PRODUCTS: ID (PK), TITLE, PRICE, DESCRIPTION, CATEGORY (FK), COMPANY**
* **ORDERS: ID (PK), TOTAL, DATE**
* **ORDER\_DETAILS: ID (PK), ORDERID (FK), PRODUCTID (FK), QUANTITY, PRICE, TOTALPRICE, DISCOUNT**

CREATE TABLE CATEGORIES(ID INT IDENTITY(1,1)PRIMARY KEY,TITLE VARCHAR(20),DESCRIPTION VARCHAR(100));

CREATE TABLE PRODUCTS(ID INT IDENTITY(1,1) PRIMARY KEY,TITLE VARCHAR(20),PRICE FLOAT,DESCRIPTION VARCHAR(50),CATEGORY INT,COMPANY VARCHAR(20),FOREIGN KEY (CATEGORY) REFERENCES CATEGORIES(ID));

CREATE TABLE ORDERS (ID INT IDENTITY(1,1)PRIMARY KEY,TOTAL FLOAT,DATE DATE);

CREATE TABLE ORDER\_DETAILS(ID INT IDENTITY(1,1)PRIMARY KEY,ORDERID INT,PRODUCTID INT,QUANTITY INT,PRICE FLOAT,TOTALPRICE AS PRICE\*QUANTITY,DISCOUNT FLOAT,FOREIGN KEY(ORDERID)REFERENCES ORDERS(ID),FOREIGN KEY(PRODUCTID) REFERENCES PRODUCTS(ID) );

* **CREATE RELATIONSHIP AS SHOWN IN THE TABLE SCHEMA**

DONE

* **ADD SOME DUMMY DATA**

INSERT INTO CATEGORIES(TITLE,DESCRIPTION) VALUES

('MOBILE PHONE','MOBILE PHONES ARE ELECTRONIC DEVICES USED FOR COMMUNICATION'),

('LED TV','BEST QUALITY LED TV FOR BEST PERFORMANCE.'),

('LAPTOP','BEST LAPTOPTS FOR EVERY USER'),

('HEADPHONES','WIRED AND WIRELESS HEADPHONES OF BEST QUALITY');

INSERT INTO PRODUCTS(TITLE,PRICE,DESCRIPTION,CATEGORY,COMPANY)VALUES

('REMI NOTE 4',13000.00,'WITH 64GB ROM AND 4GB RAM',1,'REDMI'),

('TCL FLEX 64',16000.00,'WITH 64 INCH SIZE AND HD QAALITY DISPLAY',2,'TCL'),

('HP15 AU3500U',37000.00,'WITH 1TB ROM AND 8GB RAM',3,'HP'),

('BOAT AIRDOPES 441PRO',2500.00,'5 ATM, 5 HRS BACKUP',4,'BOAT'),

('REALME ROCKERZ',1800.00,'WIRED 7 HRS BACKUP',4,'REALME'),

('LG WIDE 64',16000.00,'WITH 64 INCH SIZE AND HD QUALITY DISPLAY',2,'LG'),

('DELL ASPIRON',35000.00,'WITH 1TB ROM AND 8GB RAM',3,'DELL');

INSERT INTO ORDERS(TOTAL,DATE)VALUES

(13000.00,'2022-10-24'),

(16000.00,'2022-10-24'),

(37000.00,'2022-10-24'),

(2500.00,'2022-10-24'),

(1800.00,'2022-10-24'),

(16000.00,'2022-10-24'),

(35000.00,'2022-10-24');

INSERT INTO ORDER\_DETAILS(ORDERID,PRODUCTID,QUANTITY,PRICE,DISCOUNT)VALUES

(1,1,2,13000.00,5),

(2,2,1,16000.00,10),

(6,6,3,16000.00,5),

(4,4,1,2500.00,5),

(5,5,2,1800.00,5),

(7,7,2,35000.00,5),

(3,3,1,37000.00,5);

**ASSIGNMENT 3**

SOLVE FOLLOWING QUERIES USING SALES DATABASE

* **WRITE A QUERY THAT PRODUCES ALL ROWS FROM THE CUSTOMERS TABLE FOR WHICH THE SALESPERSON’S NUMBER IS 1001.**

SELECT \* FROM CUSTOMERS WHERE SNUM=1001;

* **WRITE A SELECT COMMAND THAT PRODUCES THE RATING FOLLOWED BY THE NAME OF EACH CUSTOMER IN SAN JOSE**

SELECT RATING,CITY,CNAME FROM CUSTOMERS WHERE CITY='SAN JOSE';

* **WRITE A QUERY THAT WILL PRODUCE THE SNUM VALUES OF ALL SALESPEOPLE FROM THE ORDERS TABLE (WITH THE DUPLICATE VALUES SUPPRESSED)**

SELECT  DISTINCT SNUM FROM SALESPEOPLE;

* **WRITE A QUERY THAT WILL GIVE YOU ALL ORDERS FOR MORE THAN RS. 1,000**

SELECT \* FROM ORDERS WHERE AMT>1000;

* **WRITE A QUERY THAT WILL GIVE YOU THE NAMES AND CITIES OF ALL SALESPEOPLE IN LONDON WITH A COMMISSION ABOVE 0.10**

SELECT SNAME,CITY FROM SALESPEOPLE WHERE CITY='LONDON' AND COMM>0.10;

* **WRITE A QUERY ON THE CUSTOMERS TABLE WHOSE OUTPUT WILL EXCLUDE ALL CUSTOMERS WITH A RATING <=100, UNLESS THEY ARE LOCATED IN ROME.**

SELECT \*FROM CUSTOMERS WHERE RATING<=100 AND CITY='ROME';

* **WHAT WILL BE THE OUTPUT FROM THE FOLLOWING QUERY?**

**SELECT \* FROM ORDERS WHERE (AMT < 1000 ORNOT (ODATE = ‘1990-10-03’AND CNUM > 2003));**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3001** | **18.69** | **1990-10-03** | **2008** | **1007** |
| **3003** | **767.19** | **1990-10-03** | **2001** | **1001** |
| **3005** | **5160.45** | **1990-10-03** | **2003** | **1002** |
| **3009** | **1713.23** | **1990-10-04** | **2002** | **1003** |
| **3007** | **75.75** | **1990-10-04** | **2004** | **1002** |
| **3008** | **4723.00** | **1990-10-04** | **2006** | **1001** |
| **3010** | **309.95** | **1990-10-04** | **2004** | **1002** |
| **3011** | **9891.88** | **1990-10-04** | **2006** | **1001** |

* **WHAT WILL BE THE OUTPUT OF THE FOLLOWING QUERY?SELECT \* FROM ORDERS WHERE NOT ((ODATE = ‘1990-10-03’ OR SNUM >1006) AND AMT >= 1500)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3001** | **18.69** | **1990-10-03** | **2008** | **1007** |
| **3003** | **767.19** | **1990-10-03** | **2001** | **1001** |
| **3006** | **1098.16** | **1990-10-03** | **2008** | **1007** |
| **3009** | **1713.23** | **1990-10-04** | **2002** | **1003** |
| **3007** | **75.75** | **1990-10-04** | **2004** | **1002** |
| **3008** | **4723.00** | **1990-10-04** | **2006** | **1001** |
| **3010** | **309.95** | **1990-10-04** | **2004** | **1002** |
| **3011** | **9891.88** | **1990-10-04** | **2006** | **1001** |

* **WRITE A QUERY THAT SELECTS ALL ORDERS EXCEPT THOSE WITH ZEROES OR NULLS IN THE AMT FIELD.**

SELECT \* FROM ORDERS WHERE (AMT IS NOT NULL AND AMT <> 0);

**ASSIGNMENT 4**

TO SOLVE BELOW QUERIES USE "HR" DATABASE

* **WRITE A QUERY TO GET UNIQUE DEPARTMENT ID FROM EMPLOYEE TABLE.**

SELECT DISTINCT DEPARTMENT\_ID FROM EMPLOYEES;

* **WRITE A QUERY TO GET ALL EMPLOYEE DETAILS FROM THE EMPLOYEE TABLE ORDER BY FIRST NAME, DESCENDING.**

SELECT \* FROM EMPLOYEES ORDER BY FIRST\_NAME DESC;

* **WRITE A QUERY TO GET THE EMPLOYEE ID, NAMES (FIRST\_NAME, LAST\_NAME), SALARY IN ASCENDING ORDER OF SALARY.**

SELECT EMPLOYEE\_ID,FIRST\_NAME+' '+LAST\_NAME AS NAMES ,SALARY FROM EMPLOYEES ORDER BY SALARY ASC;

* **DISPLAY FIRST NAME AND JOIN DATE OF THE EMPLOYEES WHO IS EITHER IT PROGRAMMER OR SALES MAN.**
* **DISPLAY DETAILS OF EMPLOYEE WITH ID 150 OR 160.**

SELECT \*FROM EMPLOYEES WHERE EMPLOYEE\_ID IN(150,160);

* **DISPLAY FIRST NAME, SALARY, COMMISSION PCT, AND HIRE DATE FOR EMPLOYEES WITH SALARY LESS THAN 10000.**

SELECT FIRST\_NAME, SALARY, COMMISSION\_PCT, HIRE\_DATE FROM EMPLOYEES WHERE SALARY<10000;

* **DISPLAY EMPLOYEES WHERE THE FIRST NAME OR LAST NAME STARTS WITH S.**

SELECT \*FROM EMPLOYEES WHERE FIRST\_NAME LIKE 'S%';

* **DISPLAY DETAILS OF JOBS IN THE DESCENDING ORDER OF THE TITLE.**

SELECT \*FROM JOBS ORDER BY JOB\_TITLE DESC;

* **DISPLAY DETAILS OF THE EMPLOYEES WHERE COMMISSION PERCENTAGE IS NULL AND SALARY IN THE RANGE 5000 TO 10000 AND DEPARTMENT IS 30.**

SELECT \* FROM EMPLOYEES WHERE COMMISSION\_PCT IS NOT NULL AND SALARY BETWEEN 5000 AND 10000 AND DEPARTMENT\_ID=30;

* **DISPLAY EMPLOYEES FIRST\_NAME,EMAIL WHO ARE WORKING IN “EXECUTIVE” DEPARTMENT.**

SELECT FIRST\_NAME, EMAIL FROM EMPLOYEES WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID FROM DEPARTMENTS WHERE DEPARTMENT\_NAME = 'EXECUTIVE');

* **DISPLAY UNIQUE CONTRY\_ID FROM LOCATIONS TABLE.**

SELECT DISTINCT COUNTRY\_ID FROM COUNTRIES;

* **DISPLAY ALL EMPLOYEES WHOSE HAVE JOB\_ID IT\_PROG AND FI\_ACCOUNT.**

SELECT \*FROM EMPLOYEES WHERE JOB\_ID IN('IT\_PROG','FI\_ACCOUNT');

* **DISPLAY ALL COUNTRIES IN ASCENDING ORDER.**

SELECT COUNTRY\_NAME FROM COUNTRIES ORDER BY COUNTRY\_NAME DESC;

**ASSIGNMENT 5**

TO SOLVE BELOW QUERIES USE "HR" DATABASE

* **DISPLAY FIRST NAME AND LAST NAME AFTER CONVERTING THE FIRST LETTER OF EACH NAME TO UPPER CASE AND THE REST TO LOWER CASE.**

SELECT UPPER(SUBSTRING(FIRST\_NAME, 1,1))+ LOWER(SUBSTRING(FIRST\_NAME, 2, LEN(FIRST\_NAME)-1))AS FIRST\_NAME ,UPPER(SUBSTRING(LAST\_NAME, 1,1))+ LOWER(SUBSTRING(LAST\_NAME, 2, LEN(LAST\_NAME)-1)) AS LAST\_NAME FROM EMPLOYEES;

* **DISPLAY THE FIRST WORD IN JOB TITLE.**

SELECT SUBSTRING(JOB\_TITLE,1,1) AS FIRSTLETTEROFJOB FROM JOBS;

* **DISPLAY THE LENGTH OF FIRST NAME FOR EMPLOYEES WHERE LAST NAME CONTAIN CHARACTER ‘B’ AFTER 3RD POSITION.**

SELECT FIRST\_NAME,LAST\_NAME,LEN(FIRST\_NAME) AS LENGTH\_OF\_FIRST\_NAME FROM EMPLOYEES WHERE (SUBSTRING(LAST\_NAME,4,LEN(LAST\_NAME))LIKE 'B%');

* **DISPLAY FIRST NAME IN UPPER CASE AND EMAIL ADDRESS IN LOWER CASE FOR EMPLOYEES WHERE THE FIRST NAME AND EMAIL ADDRESS ARE SAME IRRESPECTIVE OF THE CASE.**

SELECT UPPER(FIRST\_NAME) AS FIRST\_NAME\_UPPER,LOWER(EMAIL) AS EMAIL\_LOWER FROM EMPLOYEES WHERE (UPPER(FIRST\_NAME) LIKE LOWER(EMAIL)) OR (LOWER(FIRST\_NAME) LIKE UPPER(EMAIL));

* **DISPLAY FIRST NAME, SALARY, AND ROUND THE SALARY TO THOUSANDS.**

SELECT FIRST\_NAME, SALARY, ROUND(SALARY, -3) AS ROUNDED FROM EMPLOYEES;

* **DISPLAY EMPLOYEE ID AND THE DATE ON WHICH HE ENDED HIS PREVIOUS JOB.**

SELECT EMPLOYEE\_ID, MAX(END\_DATE) AS LAST\_DATE FROM JOB\_HISTORY GROUP BY EMPLOYEE\_ID;

* **DISPLAY FIRST NAME AND DATE OF FIRST SALARY OF THE EMPLOYEES.**

SELECT FIRST\_NAME, HIRE\_DATE, LAST\_DAY(HIRE\_DATE)+1 FROM EMPLOYEES;

* **DISPLAY FIRST NAME AND EXPERIENCE OF THE EMPLOYEES.**

SELECT EMPLOYEE\_ID, FIRST\_NAME, DEPARTMENT\_ID, JOB\_ID, CONVERT(VARCHAR(3),DATEDIFF(MONTH, HIRE\_DATE, GETDATE())/12) +' YEARS '+ CONVERT(VARCHAR(2),DATEDIFF(MONTH, HIRE\_DATE, GETDATE()) % 12)+ ' MONTHS' AS EXPERIENCE FROM EMPLOYEES;

* **DISPLAY FIRST NAME OF EMPLOYEES WHO JOINED IN 2001.**

SELECT FIRST\_NAME,HIRE\_DATE FROM EMPLOYEES WHERE HIRE\_DATE BETWEEN '2001-01-01' AND '2001-12-12';

* **DISPLAY EMPLOYEES WHO JOINED IN THE CURRENT YEAR.**

SELECT \* FROM EMPLOYEES WHERE YEAR(HIRE\_DATE) = YEAR(GETDATE());

* **DISPLAY THE NUMBER OF DAYS BETWEEN SYSTEM DATE AND 1ST JANUARY 2011.**

SELECT DATEDIFF(DAY, '20110101', GETDATE()) AS TOTAL\_DAYS;

* **DISPLAY NUMBER OF EMPLOYEES JOINED AFTER 15TH OF THE MONTH.**

SELECT \* FROM EMPLOYEES WHERE DATEPART(DAY FROM HIRE\_DATE) > 15;

* **DISPLAY THIRD HIGHEST SALARY OF EMPLOYEES.**

WITH RESULT AS (SELECT SALARY,DENSE\_RANK() OVER (ORDER BY SALARY DESC) AS DENSERANK FROM EMPLOYEES)SELECT TOP 1 SALARY FROM RESULT WHERE DENSERANK = 3;

**ASSIGNMENT 6**

TO SOLVE BELOW QUERIES USE “SALES” DATABASE

* **WRITE A QUERY THAT COUNTS THE NUMBER OF SALESPEOPLE REGISTERING ORDERS FOR EACH DAY. (IF A SALESPERSON HAS MORE THAN ONE ORDER ON A GIVEN DAY, HE OR SHE SHOULD BE COUNTED ONLY ONCE.)**

SELECT ODATE,SNUM,COUNT(\*) AS ORDERS FROM ORDERS GROUP BY ODATE,SNUM;

* **WRITE A QUERY ON THE CUSTOMERS TABLE THAT WILL FIND THE HIGHEST RATING IN EACH CITY. PUT THE OUTPUT IN THIS FORM:**

**FOR THE CITY (CITY), THE HIGHEST RATING IS : (RATING).**

SELECT CITY,MAX(RATING) MAX\_RATING FROM CUSTOMERS  GROUP BY CITY;

* **WRITE A QUERY THAT TOTALS THE ORDERS FOR EACH DAY AND PLACES THE RESULTS IN DESCENDING ORDER.**

SELECT COUNT(ODATE) "TOTAL ORDERS",ODATE FROM ORDERS GROUP BY ODATE ORDER BY "TOTAL ORDERS";

* **WRITE A QUERY THAT SELECTS THE TOTAL AMOUNT IN ORDERS FOR EACH SALESPERSON FOR WHOM THIS TOTAL IS GREATER THAN THE AMOUNT OF THE LARGEST ORDER IN THE TABLE.**

SELECT SUM(AMT)FROM ORDERS GROUP BY SNUM HAVING SUM(AMT)>(SELECT MAX(AMT) FROM ORDERS);

* **WRITE A QUERY THAT SELECTS THE HIGHEST RATING IN EACH CITY.**

SELECT CITY,MAX(RATING) AS MAXRATINGINCITY FROM CUSTOMERS GROUP BY CITY;

* **LARGEST ORDER TAKEN BY EACH SALESPERSON WITH ORDER VALUE MORE THAN RS.3000. 7. SELECT EACH CUSTOMER SMALLEST ORDER.**

SELECT ODATE, SNUM, MAX(AMT) FROM ORDERS WHERE AMT > 3000 GROUP BY ODATE, SNUM ORDER BY ODATE,SNUM;

SELECT CNUM, MIN(AMT) FROM ORDERS GROUP BY CNUM;

**ASSIGNMENT 7**

TO SOLVE BELOW QUERIES USE “SALES” DATABASE

* **WRITE A QUERY THAT LISTS EACH ORDER NUMBER FOLLOWED BY THE NAME OF THE CUSTOMER WHO MADE THE** ORDER.

SELECT ORDERS.ONUM, CUSTOMERS.CNAME

FROM ORDERS, CUSTOMERS

WHERE ORDERS.CNUM = CUSTOMERS.CNUM;

* **WRITE A QUERY THAT GIVES THE NAMES OF BOTH THE SALESPERSON AND THE CUSTOMER FOR EACH ORDER** ALONG WITH THE ORDER NUMBER.

SELECT SNAME,CNAME,ONUM

FROM ORDERS O,CUSTOMERS C,SALESPEOPLE S

WHERE O.SNUM=C.SNUM AND O.SNUM=S.SNUM;

* **WRITE A QUERY THAT PRODUCES ALL CUSTOMERS SERVICED BY SALESPEOPLE WITH A COMMISSION ABOVE 12%.OUTPUT THE CUSTOMER’S NAME, THE SALESPERSON’S NAME, AND THE SALESPERSON’S RATE OF COMMISSION.**

SELECT CNAME,

S.SNUM,

COMM\*100 "RATE OF COMM."

FROM CUSTOMERS C,

SALESPEOPLE S

WHERE C.SNUM=S.SNUM

AND COMM>.12 ;

* **WRITE A QUERY THAT CALCULATES THE AMOUNT OF THE SALESPERSON’S COMMISSION ON EACH ORDER BY A CUSTOMER WITH A RATING ABOVE 100**

SELECT SNAME, AMT \* COMM

FROM ORDERS, CUSTOMERS, SALESPEOPLE

WHERE RATING > 100 AND

SALESPEOPLE.SNUM = CUSTOMERS.SNUM AND

SALESPEOPLE.SNUM = ORDERS.SNUM AND

CUSTOMERS.CNUM = ORDERS.CNUM

* **WRITE A QUERY THAT PRODUCES ALL PAIRS OF SALESPEOPLE WHO ARE LIVING IN THE SAME CITY.EXCLUDE** **COMBINATIONS OF SALESPEOPLE WITH THEMSELVES AS WELL AS DUPLICATE ROWS WITH THE ORDER REVERSED**

SELECT M.SNAME, N.SNAME, M.CITY

FROM SALESPEOPLE M,SALESPEOPLE N

WHERE M.CITY=N.CITY

AND M.SNAME<N.SNAME;

**ASSIGNMENT 8**

TO SOLVE BELOW QUERIES USE "SALES" DATABASE

* **WRITE A QUERY THAT USES A SUBQUERY TO OBTAIN ALL ORDERS FOR THE CUSTOMER NAMED CISNEROS. ASSUME** YOU DO NOT KNOW HIS CUSTOMER NUMBER (CNUM).

SELECT ONUM "ALL ORDERS",

CNUM

FROM ORDERS

WHERE CNUM=

(SELECT CNUM

FROM CUSTOMERS

WHERE CNAME='CISNEROS');

* **WRITE A QUERY THAT PRODUCES THE NAMES AND RATINGS OF ALL CUSTOMERS WHO HAVE ABOVE- AVERAGE** ORDERS.

SELECT MAX(B.CNAME), MAX(B.RATING), A.CNUM FROM ORDERS A, CUSTOMERS B WHERE A.CNUM = B.CNUM GROUP BY A.CNUM HAVING COUNT(A.CNUM) > ( SELECT AVG(COUNT(CNUM)) FROM ORDERS GROUP BY CNUM);

* **WRITE A QUERY THAT SELECTS THE TOTAL AMOUNT IN ORDERS FOR EACH SALESPERSON FOR WHOM THIS TOTAL IS** **GREATER THAN THE AMOUNT OF THE LARGEST ORDER IN THE TABLE.**

SELECT SNUM, SUM(AMT)

FROM ORDERS

GROUP BY SNUM

HAVING SUM(AMT) > ( SELECT MAX(AMT)

FROM ORDERS);

* **WRITE A QUERY THAT SELECTS ALL CUSTOMERS WHOSE RATINGS ARE EQUAL TO OR GREATER THAN ANY OF SERIES.**

SELECT A.SNAME,B.RATING FROM SALESPEOPLE A,CUSTOMERS B WHERE A.SNUM=B.SNUM;

SELECT \* FROM CUSTOMERS WHERE RATING>=200;

* **WRITE A QUERY USING ANY OR ALL THAT WILL FIND ALL SALESPEOPLE WHO HAVE NO CUSTOMERS LOCATED IN** THEIR CITY.

SELECT A.SNAME,A.CITY,B.CNAME FROM SALESPEOPLE A,CUSTOMERS B WHERE A.CITY=B.CITY;

* **WRITE A QUERY THAT SELECTS ALL ORDERS FOR AMOUNTS GREATER THAN ANY FOR THE CUSTOMERS IN LONDON.**

SELECT \*

FROM ORDERS

WHERE AMT <

(SELECT MAX (AMT)

FROM ORDERS A, CUSTOMERS B

WHERE  A.SNUM=B.SNUM

AND B.CITY='LONDON');

* **EXTRACT ALL THE ORDERS OF MOTIKA.**

SELECT ONUM

FROM ORDERS

WHERE SNUM = ( SELECT SNUM

FROM SALESPEOPLE

WHERE SNAME = 'MOTIKA');

* **FIND ALL THE ORDER ATTRIBUTE TO SALESPEOPLE SERVICING CUSTOMERS IN LONDON.**

SELECT SNUM, CNUM

FROM ORDERS

WHERE CNUM IN (SELECT CNUM

FROM CUSTOMERS

WHERE CITY = 'LONDON');

* **FIND NAMES AND NUMBERS OF ALL SALESPERSON WHO HAVE MORE THAN ONE CUSTOMER. 10. FIND SALESPEOPLE NUMBER,NAME AND CITY WHO HAVE MULTIPLE CUSTOMERS.**

SELECT A.SNUM, A.SNAME

FROM SALESPEOPLE A

WHERE 1 <

(SELECT COUNT(\*)

FROM CUSTOMERS

WHERE A.SNUM=A.SNUM);

* **SELECT CUSTOMERS WHO HAVE A GREATER RATING THAN ANY OTHER CUSTOMER IN ROME.**

SELECT A.CNAME

FROM CUSTOMERS A

WHERE CITY = 'ROME' AND

RATING > ( SELECT MAX(RATING)

FROM CUSTOMERS

WHERE CITY != 'ROME');

* **SELECT ALL ORDERS THAT HAD AMOUNTS THAT WERE GREATER THAT AT LEAST ONE OF THE ORDERS FROM ‘1990-10-04’ .**

SELECT \*

FROM ORDERS

WHERE AMT > ANY (SELECT AMT

FROM ORDERS

WHERE ODATE = '1990-19-04');

* **FIND ALL ORDERS WITH AMOUNTS SMALLER THAN ANY AMOUNT FOR A CUSTOMER IN SAN JOSE.**

SELECT ONUM, AMT FROM ORDERS WHERE AMT < ANY ( SELECT AMT FROM ORDERS, CUSTOMERS WHERE CITY = 'SAN JOSE' AND ORDERS.CNUM = CUSTOMERS.CNUM);

* **SELECT THOSE CUSTOMERS WHOSE RATING ARE HIGHER THAN EVERY CUSTOMER IN PARIS.**

SELECT \*

FROM CUSTOMERS A

WHERE NOT EXISTS ( SELECT B.RATING

FROM CUSTOMERS B

WHERE B.CITY != 'PARIS' AND

B.RATING > A.RATING);

**ASSIGNMENT 9**

USE "SALES" DATABASE TO SOLVE BELOW QUERIES.

* **CREATE AN INDEX THAT WILL ENABLE A USER TO PULL ORDERS FOR ‘1990-10-03’ OUT OF THE ORDERS TABLE QUICKLY.**

CREATE INDEX I\_DATE ON ORDERS(ODATE);

* **IF THE ORDERS TABLE HAS ALREADY BEEN CREATED, HOW CAN YOU FORCE THE ONUM FIELD TO BE UNIQUE (ASSUME ALL CURRENT VALUES ARE UNIQUE)?**

EXEC SP\_COLUMNS ORDERS;

ALTER TABLE ORDERS

ADD CONSTRAINT UK\_ONUM UNIQUE  (ONUM);

* **CREATE AN INDEX THAT WOULD PERMIT SALESPERSON TO RETRIEVE HIS ORDERS.**

CREATE INDEX I\_DATE ON ORDERS(ODATE);

* **LET US ASSUME THAT EACH SALESPERSON IS TO HAVE ONLY ONE CUSTOMER OF A GIVEN RATING, AND THAT THIS IS CURRENTLY THE CASE. CREATE AN INDEX THAT ENFORCES IT.**

 ALTER TABLE CUSTOMERS MODIFY RATING PRIMARY KEY;

* **CREATE AN INDEX TO SPEED UP SEARCHING ORDER ON A GIVEN DATE BY GIVEN CUSTOMER**

  CREATE INDEX ODATE ON ORDERS(CNUM ASC);