# Q. Write SQL queries to perform the following based on the table PRODUCT having fields as

*(prod_id, prod_name, quantity, unit_rate, price, city)* → colnames / fields / attributes:

(i) Display those records from table PRODUCT where prod_id is more than 100.

(ii) List records from table PRODUCT where prod_name is 'Almirah'.

(iii) List all those records whose price is between 200 and 500.

(iv) Display the product names whose price is less than the average of price.

(v) Show the total number of records in the table PRODUCT.

i) Select * from PRODUCT
where prod_id > 100;

ii) Select * from PRODUCT
where prod_name = "Almirah";

iii) Select * from PRODUCT
where Price between
200 and 500;

iv) Select Prod_name
from PRODUCT
having Price < Avg(Price);

v) Select Count(*)
from PRODUCTS;

**Q. Consider the following tables STORE and SUPPLIERS. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).**

Table: Store

| Item No | Item | Scode | Qty | Rate | LastBuy |
|---|---|---|---|---|---|
| 2005 | Sharpener Classic | 23 | 60 | 8 | 31-Jun-09 |
| 2003 | Ball Pen 0.25 | 22 | 50 | 25 | 01-Feb-10 |
| 2002 | Gel Pen Premium | 21 | 150 | 12 | 24-Feb 10 |
| 2006 | Gel Pen Classic | 21 | 250 | 20 | 11-Mar-09 |
| 2001 | Eraser Small | 22 | 220 | 6 | 19-Jan-09 |
| 2004 | Eraser Big | 22 | 110 | 8 | 02-Dec-09 |
| 2009 | Ball Pen 0.5 | 21 | 180 | 18 | 03-Nov-09 |

Table: SUPPLIERS

| Scode | Sname |
|---|---|
| 21 | Premium Stationery |
| 23 | Soft Plastics |
| 23 | Tetra Supply |

i) Select * from Store ORDERBY LastBuy;

ii) Sclect itemNO, item from Store where Rate > 15;

**(i) To display details of all the items in the Store table in ascending order of LastBuy.**

**(ii) To display Itemno and item name of those from Store table whose rate is more than 15 rupees.**

(iii) To display the details of those items whose supplier code is 22 or quantity in store is more than 110 from the table Store.

Scode = 22 OR Quantity > 110 from Store.

(iv) To display minimum rate of items for each Supplier individually as per Scode from the table Store. Group by.

(v) SELECT COUNT(DISTINCT Scode) FROM STORE;

(vi) SELECT Rate*Qty FROM STORE WHERE Itemno = 2004;

Rate * Qty → expression.

(vii) SELECT Item, Sname FROM STORE S, SUPPLIER P WHERE S.Scode = P.Scode AND ItemNo = 2006;

Joining

(viii) SELECT MAX(LastBuy)FROM STORE;

iii) Select * from Store
where Scode = 22 OR Qty > 100;

vi)

| Rate_Qty |
|---|
| 880 |

v)

| Count (Distinct Scode) |
|---|
| 3 |

vii)

| Item | Sname |
|---|---|
| Gel Pen Classic | Premium Stationary |

viii)

| Max(Lastbuy) |
|---|
| 24-Feb-10 |

BOOKS

| Book-id | Book_name | Author_name | Publishers | Price | Type | Qty |
|---|---|---|---|---|---|---|
| k0001 | Let us C | Sanjay Mukherjee | EPB | 450 | Comp | 15 |
| p0001. | Genuine | J. Mukhi | FIRST PUBL. | 755 | Fiction | 24 |
| m0001 | Mastering C++ | Kantar | EPB | 165 | Comp | 60 |
| n0002 | VC++ advance | P. Purohit | TDH | 250 | Comp | 45 |
| k0002 | Programming with Python | Sanjeev | FIRST PUBL. | 350 | Fiction | 30 |

ISSUED.

| Book_ID | Qty_Issued |
|---|---|
| L02 | 13 |
| L04 | 5 |
| L05 | 21 |

INSTANCE / OBJECT.

(i) To show the books of FIRST PUBL. Publishers written by P. Purohit.
(ii) To display cost of all the books published for FIRST PUBL.
(iii) Depreciate the price of all books of EPB publishers by 5%.
(iv) To display the BOOK_NAME and price of the books, more than 3 copies of which have been issued.
(v) To show total cost of books of each type.
(vi) To show the details of the costliest book.

vi)
```
Select * from BOOKS
having Price = MAX(PRICE);
```

[H.W. All the Questions from the previous & today's class.]

Student

| RollNo | Name | Class | DOB | Gender | City | Marks |
|---|---|---|---|---|---|---|
| 1 | Nanda | X | 06-06-1995 | M | Agra | 551 |
| 2 | Saurabh | XII | 07-05-1993 | M | Mumbai | 462 |
| 3 | Sanal | XI | 06-05-1994 | F | Delhi | 400 |
| 4 | Trisla | XII | 08-08-1995 | F | Mumbai | 450 |
| 5 | Stort | XII | 08-10-1995 | M | Delhi | 369 |
| 6 | Marisla | XI | 12-12-1994 | F | Dubai | 250 |
| 7 | Neha | X | 08-12-1995 | F | Moscow | 377 |
| 8 | Nishant | X | 12-06-1995 | M | Moscow | 489 |

| COUNT | CITY |
|---|---|
| 1 | Agra |
| 2 | Delhi |
| 1 | Dubai |
| 2 | Moscow |
| 2 | Mumbai |

(i) SELECT COUNT(*), City FROM STUDENT GROUP BY CITY

(ii) SELECT MAX (DOB), MIN(DOB) FROM STUDENT;

(iii) SELECT NAME, GENDER FROM STUDENT WHERE CITY = "Delhi";

Q. Write SQL queries for (i) to (iv), which are based on the table: STUDENT given in the question 4(g):

(i) To display the records from table student in alphabetical order as per the name of the student.

(ii) To display Class, Dob and City whose marks is between 450 and 551.

(iii) To display Name, Class and total number of students who have secured more than 450 marks, class wise.

(iv) To increase marks of all students by 20 whose class is "XII".

UPDATE - SET.

iv)
```
UPDATE STUDENT
SET MARKS = MARKS + 20
where Class = "XII";
```

Gopi Krishna is using a table Employee.

It has the following columns: Code, Name, Salary, Deptcode

He wants to display maximum salary department wise.

He wrote the following command:

SELECT Deptcode, Max(Salary) FROM Employee ;

But he did not get the desired result. Rewrite the above query with necessary changes to help him get the desired output.

⇒ SELECT Deptcode, Max(salary) from EMPLOYEE

GROUP BY DeptCode;

# Q. Write SQL queries to perform the following based on the table PRODUCT having fields as

***(prod_id, prod_name, quantity, unit_rate, price, city)***

**(i) Display those records from table PRODUCT where prod_id is more than 100.**

**(ii) List records from table PRODUCT where prod_name is 'Almirah'.**

**(iii) List all those records whose price is between 200 and 500.**

**(iv) Display the product names whose price is less than the average of price.**

**(v) Show the total number of records in the table PRODUCT.**

## Q. Write SQL queries to perform the following based on the table PRODUCT having fields as

*(prod_id, prod_name, quantity, unit_rate, price, city)*

(i) Display those records from table PRODUCT where prod_id is more than 100.

(ii) List records from table PRODUCT where prod_name is 'Almirah'.

(iii) List all those records whose price is between 200 and 500.

(iv) Display the product names whose price is less than the average of price.

(v) Show the total number of records in the table PRODUCT.

iv) SELECT prod_name from PRODUCT
having price < Avg(Price);

v) SELECT count(*) from Product;

PRODUCTS TABLE

| PCODE | PNAME | COMPANY | PRICE | STOCK | MANUFACTURE | WARRANTY |
|---|---|---|---|---|---|---|
| PO01 | TV | BPL | 10000 | 200 | 12-JAN-2018 | 3 |
| POO2 | TV | SONY | 12000 | 150 | 23-MAR-2017 | 4 |
| PO03 | PC | LENOVO | 39000 | 100 | 09-APR-2018 | 2 |
| PO04 | PC | COMPAQ | 38000 | 120 | 20-JUN-2019 | 2 |
| PO05 | HANDYCAM | SONY | 18000 | 250 | 23-MAR-2017 | 3 |

(i) To show details of all PCs with stock more than 110.
(ii) To list the company which gives warranty of more than 2 years.
(iii) To find stock value of the BPL company where stock value is the sum of the products of price and stock.
(iv) To show number of products from each company.

(v) To count the number of PRODUCTS which shall be out of warranty on 20-NOV-2020.

(vi) To show the PRODUCT name of the products which are within warranty as on date.

(vii) Give the output of the following statements:
(a) Select COUNT(distinct company) from PRODUCT;
(b) Select MAX(price) from PRODUCT where WARRANTY

# School Bus

| Rtno | Area_Covered | Capacity | NoofStudents | Distance | Transporter | Charges |
|---|---|---|---|---|---|---|
| 1 | Vasant Kunj | 100 | 120 | 10 | Shivam travels | 100000 |
| 2 | Hauz Khas | 80 | 80 | 10 | Anand travels | 85000 |
| 3 | Pitampura | 60 | 55 | 30 | Anand travels | 60000 |
| 4 | Rohini | 100 | 90 | 35 | Anand travels | 100000 |
| 5 | Yamuna Vihar | 50 | 60 | 20 | Bhalla travels | 55000 |
| 6 | Krishna Nagar | 70 | 80 | 30 | Yadav travels | 80000 |
| 7 | Vasundhara | 100 | 110 | 20 | Yadav travels | 100000 |
| 8 | Paschim Vihar | 40 | 40 | 20 | Speed travels | 55000 |
| 9 | Saket | 120 | 120 | 10 | Speed travels | 100000 |
| 10 | Janakpuri | 100 | 100 | 20 | Kisan Tours | 95000 |

(vi) Give the output considering the original relation as given:
(a) Select sum(distance) from school bus where transporter= "Yadav travels";
(b) Select min(no of students) from school bus;
(c) Select avg(charges) from school bus where transporter = "Anand travels";
(d) Select distinct transporter from school bus;

(i) To show all information of students where capacity is more than the no. of students in order of rtno.

(ii) To show area_covered for buses covering more than 20 km., but charges less than 80000.

iv) To show rtno, area_covered and average cost per student for allroutes where average cost per student is --- charge / no ofstudents.

(v) Add a new record with the following data:
(11, "Motibagh", 35, 32, 10, "kisan tours", 35000)

Q. Shanaya Khanna is using a table Employee.
It has the following columns:
Admno, Name, Agg, Stream [column Agg contains Aggregate marks]

She wants to display highest Agg obtained in each Stream.

She wrote the following statement:

SELECT Stream, MAX(Agg) FROM Employee ;

But she did not get the desired result. Rewrite the above query with necessary change to help get the desired output.

→ SELECT Stream, MAX(Agg) from EMPLOYEE
GROUPBY Stream;

**Q. Write a output for SQL queries (i) to (iii), which are based on the table: STUDENT given below:**

**Table: STUDENT**

| RollNo | Name | Class | DOB | Gender | City | Marks |
|---|---|---|---|---|---|---|
| 1 | Nanda | X | 06-06-1995 | M | Agra | 551 |
| 2 | Saurabh | XII | 07-05-1993 | M | Mumbai | 462 |
| 3 | Sanal | XI | 06-05-1994 | F | Delhi | 400 |
| 4 | Trisla | XII | 08-08-1995 | F | Mumbai | 450 |
| 5 | Stort | XII | 08-10-1995 | M | Delhi | 369 |
| 6 | Marisla | XI | 12-12-1994 | F | Dubai | 250 |
| 7 | Neha | X | 08-12-1995 | F | Moscow | 377 |
| 8 | Nishant | X | 12-06-1995 | M | Moscow | 489 |

**(i) SELECT COUNT(*), City FROM STUDENT GROUP BY CITY HAVING** MARKS > 400 ;

**(ii) SELECT MAX (DOB), MIN(DOB) FROM** STUDENT ;

**(iii) SELECT NAME, GENDER FROM STUDENT WHERE CITY = "Delhi";**

i) Output:-

| Count(*) | City |
|---|---|
| 1 | Agra |
| 0 | Delhi |
| 0 | Dubai |
| 2 | Mumbai |
| 1 | Moscow. |

ii)

| Max(DOB) | Min(DOB) |
|---|---|
| 08-12-1995 | 07-05-1993 |

iii)

| Name | Gender |
|---|---|
| Sanal | F |
| Stort | M. |

**Table: FURNITURE**

| NO | ITEM | TYPE | DATEOFSTOCK | PRICE | DISCOUNT |
|---|---|---|---|---|---|
| 1 | Whitelotus | DoubleBed | 2002-02-23 | 3000 | 25 |
| 2 | Pinkfeathers | BabyCot | 2002-01-29 | 7000 | 20 |
| 3 | Dolphin | BabyCot | 2002-02-19 | 9500 | 20 |
| 4 | Decent | OfficeTable | 2002-02-01 | 25000 | 30 |
| 5 | Comfortzone | DoubleBed | 2002-02-12 | 25000 | 30 |
| 6 | Donald | Baby Cot | 2002-02-24 | 6500 | 15 |

(i) To list the details of furniture whose price is more than 10000.

(ii) To list the Item name and Price of furniture whose discount is between 10 and 20.

(iii) To delete the record of all items where discount is 30.

(iv) To display the price of 'Baby Cot'.

(v) To list item name, type and price of all items whose names start with 'D'.

(vi) Select Distinct Type from Furniture;

(vii) Select Max(Price) from Furniture where DateofStock >' 2002-02-15';

(viii) Select Count(*) from Furniture where Discount < 25;

vi) SELECT DISTINCT TYPE FROM FURNITURE;

Output:-

| TYPE |
|---|
| BabyCot |
| DoubleBed |
| OfficeTable |

vii) Output:-

| Max(Price) |
|---|
| 9500 |

viii) Output:-

| Count(*) |
|---|
| 3 |

GRADUATE

| SNO | NAME | STIPEND | SUBJECT | AVERAGE | RANK |
|---|---|---|---|---|---|
| 1 | KARAN | 400 | PHYSICS | 68 | 1 |
| 2 | RAJ | 450 | CHEMISTRY | 68 | 1 |
| 3 | DEEP | 300 | MATHS | 62 | 2 |
| 4 | DIVYA | 350 | CHEMISTRY | 63 | 1 |
| 5 | GAURAV | 500 | PHYSICS | 70 | 1 |
| 6 | MANAV | 400 | CHEMISTRY | 55 | 2 |
| 7 | VARUN | 250 | MATHS | 64 | 1 |
| 8 | LIZA | 450 | COMPUTER | 68 | 1 |
| 9 | PUJA | 500 | PHYSICS | 62 | 1 |
| 10 | NISHA | 300 | COMPUTER | 57 | 2 |

**(i) List the names of those students who have obtained rank 1 sorted by NAME**

**(ii) Display a list of all those names whose AVERAGE is greater than 65.**

**(iii) Display the names of those students who have opted COMPUTER as a subject with an AVERAGE of more than 60.**

**(iv) List the names of all the students in alphabetical order.**

**(v) SELECT * FROM GRADUATE WHERE NAME LIKE "%1%";**

**(vi) SELECT DISTINCT RANK FROM GRADUATE;**

iv) SELECT NAME FROM GRADUATE ORDER BY NAME;

v) Output:-

vi)

| DISTINCT (RANK) |
|---|
| 1 |
| 2 |

**Table: STUDENT**

| ColumnName | DataType | size | Constraint |
|---|---|---|---|
| ROLLNO | Integer | 4 | Primary Key |
| SNAME | Varchar | 25 | Not Null |
| GENDER | Char | 1 | Not Null |
| DOB | Date | | Not Null |
| FEES | Integer | 4 | Not Null |
| HOBBY | Varchar | 15 | Null |

**(i) Write SQL query to create the table.**

**(ii) Write SQL query to increase the size of SNAKE to hold 30 characters.**

**(iii) Write SQL query to remove the column HOBBY.**

**(iv) Write SQL query to insert a row in the table with any values of your choice that can be accommodated there.**

**PRODUCT:**

| P_ID | ProductName | Manufacturer | Price | Discount |
|---|---|---|---|---|
| TP01 | Talcum Powder | LAK | 40 | |
| FW05 | Face Wash | ABC | 45 | 5 |
| BS01 | Bath Soap | ABC | 55 | |
| SH06 | Shampoo | XYZ | 120 | 10 |
| FW12 | Face Wash | XYZ | 95 | |

**CLIENT:**

| C_ID | ClientName | City | P_ID |
|---|---|---|---|
| 01 | Cosmetic Shop | Delhi | TP01 |
| 02 | Total Health | Mumbai | FW05 |
| 03 | Live Life | Delhi | BS01 |
| 04 | Pretty Woman | Delhi | SHO6 |
| 05 | Dreams | Delhi | FW12 |

**(i) Write SQL Query to display Product Name and Price for all products whose Price is in the range 50 to 150.**

**(ii) Write SQL Query to display details of products whose manufacturer is either XYZ or ABC.**

**(iii) Write SQL query to display ProductName, Manufacturer and Price for all products that are not giving any discount.**

**(iv) Write SQL query to display ProductName and price for all products whose ProductName ends with 'h'.**

**(v) Write SQL query to display ClientName, City, P_ID and ProductName for all clients whose city is Delhi.**

**(vi) Which column is used as Foreign Key and name the table where it has been used as Foreign key.**

**Table: HOSPITAL**

| SNo | Name | Age | Department | Dateofadm | Charges | Sex |
|---|---|---|---|---|---|---|
| 1 | Arpit | 62 | Surgery | 21-01-98 | 300 | M |
| 2 | Zareena | 22 | ENT | 12-12-97 | 250 | F |
| 3 | Kareem | 32 | Orthopaedic | 19-02-98 | 200 | M |
| 4 | Arun | 12 | Surgery | 11-01-98 | 300 | M |
| 5 | Zubin | 30 | ENT | 12-01-98 | 250 | M |
| 6 | Ketaki | 16 | ENT | 24-02-98 | 250 | F |
| 7 | Ankit | 29 | Cardiology | 20-02-98 | 800 | F |
| 8 | Zareen | 45 | Gynaecology | 22-02-98 | 300 | F |
| 9 | Kush | 19 | Cardiology | 13-01-98 | 800 | M |
| 10 | Shilpa | 23 | Nuclear Medicine | 21-02-98 | 400 | F |

**(a) To list the names of all the patients admitted after 15/01/98.**

**(b) To list the names of female patients who are in ENT department.**

**(c) To list the names of all patients with their date of admission in ascending order.**

**(d) To display Patient's Name, Charges, Age for only female patients.**

**(e) Find out the output of the following SQL commands:**

**(i) Select COUNT(DISTINCT charges) from HOSPITAL;**

**(ii) Select MIN(Age) from HOSPITAL where Sex = "F";**