# Database Management systems
# LABORATORY FILE

Submitted to: Dr. Inderjit Singh Dhanoa

Submitted by: Guneet Kohli

D3 CSE A1

URN: 1805172

CRN: 1815017

Guneet Kohli 1805172

# PRACTICAL-1

## Write a query for data definition (create, drop, alter and rename) and data manipulation language (Select,Insert,Update and Delete)

DATA DEFINITION LANGUAGE:

- CREATE

SQL Statement:

```
create table student(roll no varchar(10),Fname varchar(20),Lname varchar(20),marks varchar(10));
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database.

- RENAME

SQL Statement:

```
alter table student rename to college_student;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database.

- ALTER

SQL Statement:

```
ALTER TABLE COLLEGE_STUDENT
ADD COLUMN GRADE VARCHAR(5);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database.

- DROP

SQL Statement:

```
DROP TABLE COLLEGE_STUDENT;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database.

DATA MANIPULATION LANGUAGE:

- INSERT

SQL Statement:

```
insert into student values('1','ram','kishan','40');
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database. Rows affected: 1

- UPDATE

```
update student set marks='55'where fname='ram';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 3

| roll | Fname | Lname | marks |
|---|---|---|---|
| 1 | ram | kishan | 40 |
| 2 | mohan | raj | 70 |
| 3 | sarabjeet | singh | 75 |

- DELETE

SQL Statement:

```
delete from college_student where marks='55';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

You have made changes to the database. Rows affected: 1

# PRACTICAL-2

## Write SQL queries using logical operators.

### SQL Statement:

```
select shippername from shippers where salary=1000;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

Number of Records: 1

| ShipperName |
| --- |
| Speedy Express |

### SQL Statement:

```
select shippername from shippers where salary<500;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

Number of Records: 2

| ShipperName |
| --- |
| Speedy Express |
| United Package |

# PRACTICAL-3

## Write SQL queries using SQL operators(between,and,or,in ,like ,null)

- Null

SQL Statement:

```
select shippername from shippers where salary is null;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| ShipperName |
| --- |
| Federal Shipping |

- Between

SQL Statement:

```
select shippername from shippers where salary between 1000 and 2000;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 2

| ShipperName |
| --- |
| Speedy Express |
| United Package |

- Like

```
select shipperid from shippers where shippername like's%s'
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| ShipperID |
| --- |
| 1 |

- OR

SQL Statement:

```
select shippername from shippers where shipperid=1 or shipperid=2;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 2

| ShipperName |
| --- |
| Speedy Express |
| United Package |

- AND

SQL Statement:

```
select shipperid from shippers where salary is null and shipperid<=3;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| ShipperID |
| --- |
| 3 |

# PRACTICAL-4

Write SQL query using character, number, date and group functions.

- Character functions

SQL Statement:

```
SELECT lower(shippername) from Shippers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 3

| lower(shippername) |
| --- |
| speedy express |
| united package |
| federal shipping |

SQL Statement:

```
SELECT upper(shippername) from Shippers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 3

| upper(shippername) |
| --- |
| SPEEDY EXPRESS |
| UNITED PACKAGE |
| FEDERAL SHIPPING |

SQL Statement:

```
SELECT REPLACE("SQL Tutorial", "SQL", "HTML");
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| REPLACE("SQL Tutorial", "SQL", "HTML") |
| --- |
| HTML Tutorial |

```
1  SELECT TRIM(LEADING 'S' FROM cust_name )
2  FROM customer;
```

Output:

```
TRIM(LEADING'S'FROMCUST_NAME)
--------------------------------
Holmes
Micheal
Albert
Ravindran
Cook
tuart
Bolt
Fleming
Jacks
Yearannaidu
asikant
Ramanathan
Avinash
Winston
Karl
```

SQL Statement:

```
SELECT SUBSTR("SQL Tutorial", 5, 3) AS ExtractString;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| ExtractString |
| --- |
| Tut |

- Date functions

SQL Statement:

```
SELECT CURRENT_DATE();
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| CURRENT_DATE() |
| --- |
| 2020-12-22 |

SQL Statement:

```
SELECT CURRENT_TIME();
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| CURRENT_TIME() |
| --- |
| 15:28:25 |

SQL Statement:

```
SELECT YEAR("2017-06-15");
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| YEAR("2017-06-15") |
| --- |
| 2017 |

SQL Statement:

```
SELECT CURDATE();
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| CURDATE() |
| --- |
| 2020-12-22 |

- Number functions

SQL Statement:

```
SELECT TRUNCATE(135.375, 2);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| TRUNCATE(135.375, 2) |
| --- |
| 135.37 |

SQL Statement:

```
SELECT ROUND(135.375, 2);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| ROUND(135.375, 2) |
| --- |
| 135.38 |

SQL Statement:

```
SELECT POWER(4, 2);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| POWER(4, 2) |
| --- |
| 16 |

SQL Statement:

```
SELECT SUM(Quantity) AS TotalItemsOrdered FROM OrderDetails;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| TotalItemsOrdered |
|---|
| 51317 |

- Group functions

SQL Statement:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 21

| COUNT(CustomerID) | Country |
|---|---|
| 3 | Argentina |
| 2 | Austria |
| 2 | Belgium |
| 9 | Brazil |
| 3 | Canada |

SQL Statement:

```
SELECT Shippers.ShipperName,COUNT(Orders.OrderID) AS NumberOfOrders FROM Orders
LEFT JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY ShipperName;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 3

| ShipperName | NumberOfOrders |
|---|---|
| Federal Shipping | 68 |
| Speedy Express | 54 |
| United Package | 74 |

# PRACTICAL-5

## Write SQL for relational algebra (union,intersect and minus)

- Union

SQL Statement:

```
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 94

| City |
|---|
| Aachen |
| Albuquerque |
| Anchorage |
| Ann Arbor |

- Intersect

```
SELECT  ID, NAME, Amount, Date
     FROM Customers
     LEFT JOIN Orders
     ON Customers.ID = Orders.Customer_id
INTERSECT
     SELECT  ID, NAME, Amount, Date
     FROM Customers
     RIGHT JOIN Orders
     ON Customers.ID = Orders.Customer_id;
```

Output:

| ID | Name | Amount | Date |
|---|---|---|---|
| 3 | Akash | 3000 | 2017-10-08 |
| 3 | Akash | 1500 | 2017-10-08 |
| 2 | Pratik | 1560 | 2017-11-20 |
| 4 | Varun | 2060 | 2016-05-20 |

- Minus

```
SELECT NAME, AGE , GRADE
FROM Table1
MINUS
SELECT NAME, AGE, GRADE
FROM Table2
```

Output:

| Name | Age | Grade |
|---|---|---|
| Harsh | 20 | A |
| Gaurav | 21 | B |
| Pratik | 21 | A |

# PRACTICAL-6

Write SQL Queries for extracting data from more than one table(equi join,non equi join,outer join)

- Equi join

```
SELECT student.name, student.id, record.class, record.city
FROM student
JOIN record
ON student.city = record.city;
```

**Output :**

| NAME | ID | CLASS | CITY |
|---|---|---|---|
| Hina | 3 | 3 | Delhi |
| Megha | 4 | 3 | Delhi |
| Gouri | 6 | 3 | Delhi |
| Hina | 3 | 2 | Delhi |
| Megha | 4 | 2 | Delhi |
| Gouri | 6 | 2 | Delhi |
| Hina | 3 | 2 | Delhi |
| Megha | 4 | 2 | Delhi |
| Gouri | 6 | 2 | Delhi |

- Non –equi join

```
SELECT student.name, record.id, record.city
FROM student, record
WHERE Student.id < Record.id ;
```

**Output :**

| NAME | ID | CITY |
|---|---|---|
| Hina | 9 | Delhi |
| Megha | 9 | Delhi |
| Gouri | 9 | Delhi |
| Hina | 10 | Delhi |
| Megha | 10 | Delhi |
| Gouri | 10 | Delhi |
| Hina | 12 | Delhi |
| Megha | 12 | Delhi |
| Gouri | 12 | Delhi |

- Outer join

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

A selection from the result set may look like this:

| CustomerName | OrderID |
|---|---|
| Alfreds Futterkiste | *Null* |
| Ana Trujillo Emparedados y helados | 10308 |
| Antonio Moreno Taquería | 10365 |

# PRACTICAL-7

## Write SQL queries for sub queries, nested queries.

```
SELECT job_id,AVG(salary)
SELECT job_id,AVG(salary)
FROM employees
GROUP BY job_id
HAVING AVG(salary)<
(SELECT MAX(myavg) from (select job_id,AVG(min_salary) as myavg
FROM jobs
WHERE job_id IN
(SELECT job_id FROM job_history
WHERE department_id
BETWEEN 50 AND 100)
GROUP BY job_id) ss);
```

*The above code is executed in PostgreSQL 9.3*

Output

```
JOB_ID     AVG(SALARY)
---------- -----------
IT_PROG           5760
AC_ACCOUNT        8300
ST_MAN            7280
AD_ASST           4400
SH_CLERK          3215
FI_ACCOUNT        7920
PU_CLERK          2780
SA_REP            8350
```

- Sub query

```
SELECT MAX(AVG(min_salary))
FROM jobs
WHERE job_id
IN(
'ST_CLERK','ST_CLERK','IT_PROG',
'SA_REP','SA_MAN','AD_ASST',     '
AC_ACCOUNT')
GROUP BY job_id;
```

Output:

```
MAX(AVG(MIN_SALARY))
--------------------
               10000
```

```
SELECT OUTSTANDING_AMT
FROM CUSTOMER
WHERE GRADE=3
AND CUST_COUNTRY<>'India'
AND opening_amt<7000
AND EXISTS(
SELECT *
FROM agents
WHERE commission<.12);
```

Output:

```
OUTSTANDING_AMT
---------------
           6000
           3000
           5000
```

# PRACTICAL-8

## Queries (along with sub queries) using any,all,in,exists,non exists,union constraints.

SQL Statement:

```
SELECT ProductName
FROM Products
WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 31

| ProductName |
|---|
| Chais |
| Chang |
| Chef Anton's Cajun Seasoning |
| Uncle Bob's Organic Dried Pears |

SQL Statement:

```
SELECT ProductName
FROM Products
WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 0

| ProductName |
|---|

SQL Statement:

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price = 22);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| SupplierName |
|---|
| New Orleans Cajun Delights |

## SQL Statement:

```
SELECT SupplierName
FROM Suppliers
WHERE EXISTS (SELECT ProductName FROM Products WHERE Products.SupplierID = Suppliers.supplierID AND Price < 20);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 24

| SupplierName |
|---|
| Exotic Liquid |
| New Orleans Cajun Delights |
| Tokyo Traders |
| Mayumi's |
| Pavlova, Ltd. |

## SQL Statement:

```
SELECT * FROM Customers
WHERE Country IN ('Germany', 'France', 'UK');
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 29

| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
|---|---|---|---|---|---|---|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 6 | Blauer See Delikatessen | Hanna Moos | Forsterstr. 57 | Mannheim | 68306 | Germany |
| 7 | Blondel père et fils | Frédérique Citeaux | 24, place Kléber | Strasbourg | 67000 | France |
| 9 | Bon app' | Laurence Lebihans | 12, rue des Bouchers | Marseille | 13008 | France |

# PRACTICAL-9

## Queries using aggregate functions(count,sum,avg,min,max),group by,having and creation and dropping of views.

SQL Statement:

```
SELECT MAX(Price) AS LargestPrice
FROM Products;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| LargestPrice |
| --- |
| 263.5 |

SQL Statement:

```
SELECT MIN(Price) AS SmallestPrice
FROM Products;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| SmallestPrice |
| --- |
| 2.5 |

SQL Statement:

```
SELECT SUM(Quantity)
FROM OrderDetails;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| SUM(Quantity) |
| --- |
| 12743 |

## SQL Statement:

```
SELECT AVG(Price)
FROM Products;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 1

| AVG(Price) |
| --- |
| 28.866363636363637 |

## SQL Statement:

```
SELECT COUNT(ProductID)
FROM Products;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 1

| COUNT(ProductID) |
| --- |
| 77 |

## SQL Statement:

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5
ORDER BY COUNT(CustomerID) DESC;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 5

| COUNT(CustomerID) | Country |
| --- | --- |
| 13 | USA |
| 11 | Germany |
| 11 | France |
| 9 | Brazil |
| 7 | UK |

## SQL Statement:

```
CREATE VIEW [Brazil Customers] AS
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = 'Brazil';
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

You have made changes to the database.

## SQL Statement:

```
DROP VIEW [Brazil Customers];
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

You have made changes to the database.

# PRACTICAL-10

Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

SQL Statement:

```
SELECT CONVERT(varchar, '2017-08-25', 101);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| |
|---|
| 2017-08-25 |

SQL Statement:

```
SELECT CONVERT(datetime, '2017-08-25');
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| |
|---|
| 2017-08-25 00:00:00.000 |

SQL Statement:

```
SELECT CONVERT(varchar, 25.65);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

| |
|---|
| 25.65 |

**To_char fxn:**

select to_char(65,'RN')from dual;

**Output:**

LXV

**To_number fxn :**

Select to_number('1234.64') from Dual;

**Output:**

1234.64

**To_date fxn:**

SELECT TO_DATE('January 15, 1989, 11:00 A.M.')FROM DUAL

**Output:**

TO_DATE('-------- 15-JAN-89

**String functions:**

**Concat fxn:**

SELECT CONCAT('ORACLE','CORPORATION')FROM DUAL;

**Output:**

ORACLECORPORATION

**Lpad fxn:**

SELECT LPAD('ORACLE',15,'*')FROM DUAL;

**Output:**

*********ORACLE

**Rpad fxn:**

SELECT RPAD ('ORACLE',15,'*')FROM DUAL;

**Output:**

ORACLE*********

**Ltrim fxn:**

SELECT LTRIM('SSMITHSS','S')FROM DUAL;

**Output:**

MITHSS

**Rtrim fxn:**

SELECT RTRIM('SSMITHSS','S')FROM DUAL;

**Output:**

SSMITH

**Lower fxn:**

SELECT LOWER('DBMS')FROM DUAL;

**Output:**

dbms

**Upper fxn:**

SELECT UPPER('dbms')FROM DUAL;

**Output:**

DBMS

**Length fxn:** .

SELECT LENGTH('DATABASE')FROM DUAL;

**Output:**

8

**Substr fxn:**

SELECT SUBSTR('ABCDEFGHIJ'3,4)FROM DUAL;

**Output:**

CDEF

**Instr fxn:**

SELECT INSTR('CORPORATE FLOOR','OR',3,2)FROM DUAL;

**Output:**

14

**Date functions:**

**Sysdate fxn:**

SELECT SYSDATE FROM DUAL;

**Output:**

29-DEC-08

**next_day fxn:**

SQL>SELECT NEXT_DAY(SYSDATE,'WED')FROM DUAL;

**Output:**

05-JAN-09

**add_months fxn:**

SELECT ADD_MONTHS(SYSDATE,2)FROM DUAL;

**Output:**

28-FEB-09

**last_day fxn:**

SELECT LAST_DAY(SYSDATE)FROM DUAL;

**Output:**

31-DEC-08

**months_between fxn:**

SELECT MONTHS_BETWEEN(SYSDATE,HIREDATE)FROM EMP;

**Output:**

4

**Least fxn:**

SELECT LEAST('10-JAN-07','12-OCT-07')FROM DUAL;

10-JAN-07

**Greatest fxn:**

SELECT GREATEST('10-JAN-07','12-OCT-07')FROM DUAL;

**Output:**

10-JAN-07

**Trunc fxn:**

SELECT TRUNC(SYSDATE,'DAY')FROM DUAL;

**Output:**

28-DEC-08

**Round fxn:**

SELECT ROUND(SYSDATE,'DAY')FROM DUAL;

**Output:**

28-DEC-08

**to_char fxn:**

select to_char(sysdate, "dd\mm\yy") from dual;

**Output:**

24-mar-05.

**to_date fxn:**

select to_date(sysdate, "dd\mm\yy") from dual;

**Output:**

24-mar-05.

# PRACTICAL-11

## Write SQL queries to create views and also apply different operations on views.

### SQL Statement:

```
CREATE VIEW [Products Above Average Price] AS
SELECT ProductName, Price
FROM Products
WHERE Price > (SELECT AVG(Price) FROM Products);
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

You have made changes to the database.

### SQL Statement:

```
SELECT * FROM [Products Above Average Price];
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

Number of Records: 25

| ProductName | Price |
|---|---|
| Uncle Bob's Organic Dried Pears | 30 |
| Northwoods Cranberry Sauce | 40 |
| Mishi Kobe Niku | 97 |
| Ikura | 31 |
| Queso Manchego La Pastora | 38 |
| Alice Mutton | 39 |

### SQL Statement:

```
DROP VIEW [Brazil Customers];
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

### Result:

You have made changes to the database.

Result:

You have made changes to the database.