
	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

S.No.	Experiment	Issue Date	Page No.
1	Write the queries for Data definition and data manipulation language.		
2	Write SQL queries using Logical operators.		
3	Write SQL queries using SQL operators (Between.... AND, IN(List), Like, ISNULL and also with negative expressions).		
4	Write SQL queries using character, number, date and group functions.		
5	Write SQL queries for Relational Algebra(UNION, INTERSECT, MINUS, etc.)		
6	Write SQL queries for extracting data from more than one table (Equi join, Non-equi-join, Outer join).		
7	Write SQL queries for sub queries and nested queries.		
8	Write program by using PL/SQL.		
9	Concepts of ROLL BACK, COMMIT and CHECK POINTS.		
10	Create Views, Cursors and triggers and write assertions.		
11	Create FORMS and REPORTS		

### **LAB EXPERIMENT NO :1**

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

**OBJECTIVE:** Write the queries for Data definition and Data Manipulation Language.

### **SQL DML and DDL**

SQL can be divided into two parts: The Data Manipulation Language (DML) and the Data Definition Language (DDL).

The query and update commands form the DML part of SQL:

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database

The DDL part of SQL permits database tables to be created or deleted. It also define indexes (keys), specify links between tables, and impose constraints between tables. The most important DDL statements in SQL are:

- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index


### **CREATE DATABASE**

Create database database name

Eg : create database rahul

### **The CREATE TABLE Statement**

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

The CREATE TABLE statement is used to create a table in a database.

### SQL CREATE TABLE Syntax

```
CREATE TABLE table_name
(
column_name1 data_type,
column_name2 data_type,
column_name3 data_type,
....
)
```

### CREATE TABLE Example

```
CREATE TABLE Persons
(
P_Id int,
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)
```

### The INSERT INTO Statement

The INSERT INTO statement is used to insert a new row in a table.

### SQL INSERT INTO Syntax


It is possible to write the INSERT INTO statement in two forms.

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
OR
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

### SQL INSERT INTO Example

```
INSERT INTO Persons
```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

VALUES (4,'Nilsen', 'Johan', 'Bakken 2', 'Stavanger')

### **The SQL SELECT Statement**

The SELECT statement is used to select data from a database. The result is stored in a result table, called the result-set.

#### **SQL SELECT Syntax**

```
SELECT column_name(s)
FROM table_name
```

and

```
SELECT * FROM table_name
```

#### **An SQL SELECT Example**

```
SELECT LastName,FirstName FROM Persons
```

```
SELECT * Example
```

```
SELECT * FROM Persons
```

### **The UPDATE Statement**

The UPDATE statement is used to update existing records in a table.


#### **SQL UPDATE Syntax**

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

#### **SQL UPDATE Example**

```
UPDATE Persons
SET Address='Nissestien 67', City='Sandnes'
WHERE LastName='Tjessem' AND FirstName='Jakob'
```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University, Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### **The SQL SELECT Statement**

The SELECT statement is used to select data from a database. The result is stored in a result table, called the result-set.

#### **SQL SELECT Syntax**

```
SELECT column_name(s)
FROM table_name
```

And

```
SELECT * FROM table_name
```

#### **An SQL SELECT Example**

```
SELECT LastName,FirstName FROM Persons
```

#### **SELECT \* Example**

```
SELECT * FROM Persons
```

### **The DELETE Statement**

The DELETE statement is used to delete rows in a table.


#### **SQL DELETE Syntax**

```
DELETE FROM table_name
WHERE some_column=some_value
```

#### **SQL DELETE Example**

```
DELETE FROM Persons
WHERE LastName='Tjessel' AND FirstName='Jakob'
```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### **The ALTER TABLE Statement**

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

#### **SQL ALTER TABLE Syntax**

```
ALTER TABLE table_name
ADD column_name datatype
```

#### **SQL ALTER TABLE Example**

```
ALTER TABLE Persons
ADD DateOfBirth date
```

### **The DROP TABLE Statement**

The DROP TABLE statement is used to delete a table.  
 DROP TABLE table\_name

### **SQL CREATE INDEX Syntax**

Creates an index on a table. Duplicate values are allowed:


```
CREATE INDEX index_name
ON table_name (column_name)
```

#### **CREATE INDEX Example**

```
CREATE INDEX PIndex
ON Persons (LastName)
```

### **The DROP INDEX Statement**

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

The DROP INDEX statement is used to delete an index in a table.

DROP INDEX table\_name.index\_name

### **Assignment Problems:**

1. Create a table named ‘employees’ which has following fields:

	Field-name	Type	Constraints
(i)	id	INTEGER	PRIMARY KEY
(ii)	first_name	CHAR(50)	NULL
(iii)	last_name	CHAR(75)	NOT NULL
(iv)	dateofbirth	DATE	NULL

2. Insert following values into the ‘employee’ table:

id	first_name	last_name	dateofbirth
1	Trilok	Sharma	11 dec, 1974
2	Krishna	Mishra	23 june,1978
3	Gopal	Shah	12 jan, 1977
4	Kulmohan	Tiwari	16 Mar, 1980
5	Shiv	Goyal	26 Apr, 1979

3. Show all the fields of the above table.


4. Show only those employee whose last name is ‘Mishra’.

5. Update the record of employee named ‘Gopal’ with dateofbirth 17 Mar, 1978.

6. Delete the record of Employee names ‘Shiv’.

7. Add the new column named ‘dateofjoining’ in the above table.

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University, Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

8. Create an index on the above table.
9. Delete the index which is created on the above section.
10. Delete the entire table.

## **LAB EXPERIMENT NO :2**

**OBJECTIVE :** Write SQL queries using Logical operators.

### **The AND & OR Operators**

The AND operator displays a record if both the first condition and the second condition is true.

The OR operator displays a record if either the first condition or the second condition is true.

#### **AND Operator Example**

```
SELECT * FROM Persons
WHERE FirstName='Tove'
AND LastName='Svendson'
```


#### **OR Operator Example**

```
SELECT * FROM Persons
WHERE FirstName='Tove'
OR FirstName='Ola'
```

### **Combining AND & OR**

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---



	<b>Uttarakhand Open University, Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

```

SELECT * FROM Persons WHERE
LastName='Svendson'
AND (FirstName='Tove' OR FirstName='Ola')

```

### **"NOT" Logical Operator:**

If you want to find rows that do not satisfy a condition, you can use the logical operator, NOT. NOT results in the reverse of a condition. That is, if a condition is satisfied, then the row is not returned.


**For example:** If you want to find out the names of the students who do not play football, the query would be like:

```

SELECT first_name, last_name, address
FROM person
WHERE NOT address = 'dehradun'

```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### **Assignment Problems:**

Consider the following table:


The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Harry	Porter	II street	London
2	perry	Tom	Borum 20	Sydney
3	Pettersen	Tarry	Stare 20	Los Angles

1. Select only the persons with the first name equal to "Tom" AND the last name equal to "Perry".
2. Select only the persons with the first name equal to "Tom" OR the first name equal to "Tarry".
3. Select only the persons with the last name equal to "Harry" AND the first name equal to "Porter" OR to "Tom".
4. Show only those persons who are not lived in 'London'.

### **LAB EXPERIMENT NO :3**

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

**OBJECTIVE:** Write SQL queries using SQL operators (Between.... AND, IN(List), Like, ISNULL and also with negative expressions).

### **The BETWEEN Operator:**

The BETWEEN operator selects a range of data between two values. The values can be numbers, text, or dates.

#### **SQL BETWEEN Syntax**

```
SELECT column_name(s)
FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

#### **BETWEEN Operator Example**

```
SELECT * FROM Persons
WHERE LastName
BETWEEN 'Hansen' AND 'Pettersen'
```

### **The IN Operator**

The IN operator allows you to specify multiple values in a WHERE clause.

#### **SQL IN Syntax**


```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,value2,...)
```

#### **IN Operator Example**

```
SELECT * FROM Persons
WHERE LastName IN ('Hansen','Pettersen')
```

### **The LIKE Operator**

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

The LIKE operator is used to search for a specified pattern in a column.

### **SQL LIKE Syntax**

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern
```

### **LIKE Operator Example**

```
SELECT * FROM Persons
WHERE City LIKE 's%'
```

### **SQL ISNULL() Function:**


```
SELECT LastName,FirstName,Address FROM Persons
WHERE Address IS NULL
```

### **Assignment Problem**

Consider the following Table:

<b>Employee</b>	<b>Date</b>	<b>Hours</b>
John Smith	5/6/2004	8
Allan Babel	5/6/2004	8
Tina Crown	5/6/2004	8
John Smith	5/7/2004	9
Allan Babel	5/7/2004	8
Tina Crown	5/7/2004	10
John Smith	5/8/2004	8

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

Allan Babel	5/8/2004	8
Tina Crown	5/8/2004	9

- (1) Select only the entries where the column Date has value of '5/6/2004' or '5/7/2004'.
- (2) select all Employee having Name starting with 'J'.
- (3) select all Customers having 'Hours' column between '9' and '10'.
- (4) Select those employee whose hours is null.

## **LAB EXPERIMENT NO. :4**

**OBJECTIVE:** Write SQL queries using character, number, date and group functions.


### **SQL Dates**

The most difficult part when working with dates is to be sure that the format of the date you are trying to insert, matches the format of the date column in the database.

The following table lists the most important built-in date functions in MySQL:

Function	Description
----------	-------------

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

NOW()	Returns the current date and time
CURDATE()	Returns the current date
CURTIME()	Returns the current time
DATE()	Extracts the date part of a date or date/time expression
EXTRACT()	Returns a single part of a date/time
DATE_ADD()	Adds a specified time interval to a date
DATE_SUB()	Subtracts a specified time interval from a date
DATEDIFF()	Returns the number of days between two dates
DATE_FORMAT()	Displays date/time data in different formats

### **SQL Date Data Types**


SQL comes with the following data types for storing a date or a date/time value in the database:

- DATE - format YYYY-MM-DD
- DATETIME - format: YYYY-MM-DD HH:MM:SS
- TIMESTAMP - format: YYYY-MM-DD HH:MM:SS
- YEAR - format YYYY or YY

### **SQL Group Functions**

- AVG() - Returns the average value
- COUNT() - Returns the number of rows
- FIRST() - Returns the first value
- LAST() - Returns the last value
- MAX() - Returns the largest value
- MIN() - Returns the smallest value
- SUM() - Returns the sum

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

- UCASE() - Converts a field to upper case
- LCASE() - Converts a field to lower case
- MID() - Extract characters from a text field
- LEN() - Returns the length of a text field
- ROUND() - Rounds a numeric field to the number of decimals specified
- NOW() - Returns the current system date and time
- FORMAT() - Formats how a field is to be displayed

### **SQL AVG() Example**

We have the following "Orders" table:

O_Id	OrderDate	OrderPrice	Customer
1	2008/11/12	1000	Hansen
2	2008/10/23	1600	Nilsen
3	2008/09/02	700	Hansen
4	2008/09/03	300	Hansen
5	2008/08/30	2000	Jensen
6	2008/10/04	100	Nilsen

Now we want to find the average value of the "OrderPrice" fields.

We use the following SQL statement:


```
SELECT AVG(OrderPrice) AS OrderAverage FROM Orders
```

### **SQL COUNT(column\_name) Syntax**

```
SELECT COUNT(*) FROM table_name
```

### **SQL FIRST() Syntax**

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

SELECT FIRST(OrderPrice) AS FirstOrderPrice FROM Orders

### SQL LAST() Syntax

SELECT LAST(column\_name) FROM table\_name

### SQL MAX() Syntax

SELECT MAX(column\_name) FROM table\_name

### SQL MIN() Syntax

SELECT MIN(column\_name) FROM table\_name

### SQL SUM() Syntax

SELECT SUM(column\_name) FROM table\_name

### SQL UCASE() Syntax

SELECT UCASE(column\_name) FROM table\_name

### SQL LCASE() Syntax

SELECT LCASE(column\_name) FROM table\_name

### SQL MID() Syntax

SELECT MID(column\_name,start[,length]) FROM table\_name

### SQL ROUND() Syntax

SELECT ROUND(column\_name,decimals) FROM table\_name


### Assignment Problems:

Consider the following table named 'Employee':

Employee	Date	Hours
John Smith	5/6/2004	8

**PREAPEARED BY: Balam Singh Dafouti**




	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
<b>DEPTT.: Computer</b> <b>Science &amp; IT</b>	<b>LABORATORY: Database Management System</b> <b>(BCA-09)</b>	<b>SEMESTER: III</b>

Allan Babel	5/6/2004	8
Tina Crown	5/6/2004	8
John Smith	5/7/2004	9
Allan Babel	5/7/2004	8
Tina Crown	5/7/2004	10
John Smith	5/8/2004	8
Allan Babel	5/8/2004	8

- (1) Count the number of employees in our above table.
- (2) Find the name of the employee who has maximum hours.
- (3) Find the name of the employees who has minimum hours.
- (4) Select the average value for 'Hours'.
- (5) Select the total for 'Hours'.
- (6) Display Date in the format 'DD-Month-YY'.
- (7) Show the name of employees in the upper case.

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

## **LAB EXPERIMENT NO :5**

**Objective:** Write SQL queries for Relational Algebra (UNION, INTERSECT, MINUS, etc.)

### **The SQL UNION Operator**

The UNION operator is used to combine the result-set of two or more SELECT statements.

### **SQL UNION Syntax**

```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2
```


### **SQL UNION ALL Syntax**

```
SELECT column_name(s) FROM table_name1
UNION ALL
SELECT column_name(s) FROM table_name2
```

### **SQL INTERSECT Syntax**

```
SELECT Date FROM Store_Information
INTERSECT SELECT Date FROM Internet_Sales
```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### SQL Minus

```
SELECT Date FROM Store_Information
MINUS
SELECT Date FROM Internet_Sales
```

### SQL LIMIT

```
SELECT store_name, Sales, Date
FROM Store_Information
ORDER BY Sales DESC
LIMIT 2;
```

### The TOP Clause

```
SELECT TOP number|percent column_name(s)
FROM table_name
```

### Assignment Problem:

Consider the following two table:


(a) "Employees\_Norway":

E_ID	E_Name
01	Hansen, Ola
02	Svendson, Tove
03	Svendson, Stephen
04	Pettersen, Kari

(b) "Employees\_USA":

E_ID	E_Name
------	--------

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III


01	Turner, Sally
02	Kent, Clark
03	Svendson, Stephen
04	Scott, Stephen

- (1) list all the different employees in Norway and USA.
- (2) list only employees of Norway.
- (3) list only employees of USA
- (4) List employees who are in both 'Norway' & 'USA'.
- (5) List the first two employees who are come first in the alphabetical order.

## **LAB EXPERIMENT NO:6**

**OBJECTIVE:** Write SQL queries for extracting data from more than one table.

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

## SQL JOIN

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables. Tables in a database are often related to each other with keys.

A primary key is a column (or a combination of columns) with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

SQL Joins can be classified into Equi join and Non Equi join.

### 1) SQL Equi joins

It is a simple sql join condition which uses the equal sign as the comparison operator. Two types of equi joins are SQL Outer join and SQL Inner join.

**For example:** You can get the information about a customer who purchased a product and the quantity of product.

### 2) SQL Non equi joins


It is a sql join condition which makes use of some comparison operator other than the equal sign like >, <, >=, <=

### SQL INNER JOIN Syntax

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

### SQL RIGHT JOIN Syntax

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

```

SELECT column_name(s)
FROM table_name1
RIGHT JOIN table_name2
ON table_name1.column_name=table_name2.column_name

```

### **SQL LEFT JOIN Syntax**

```

SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name

```

### **SQL FULL JOIN Syntax**

```

SELECT column_name(s)
FROM table_name1
FULL JOIN table_name2
ON table_name1.column_name=table_name2.column_name


```

### **Assignment Problem:**

#### **Customers:**

CustomerID	FirstName	LastName	Email	DOB	Phone
1	John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222
2	Steven	Goldfish	goldfish@fishhere.net	4/4/1974	323 455-4545
3	Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

4	James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888
---	-------	-------	--------------------	------------	--------------

Sales:

CustomerID	Date	SaleAmount
2	5/6/2004	\$100.22
1	5/7/2004	\$99.95
3	5/7/2004	\$122.95
3	5/13/2004	\$100.00
4	5/22/2004	\$555.55

- (1) Select all distinct customers (their first and last names) and the total respective amount of dollars they have spent.
- (2) display all the customers and their sales, no matter if they have ordered something or not.
- (3) List the customers in the Customers table, which still hasn't made any orders.

### LAB EXPERIMENT NO :7


**OBJECTIVE:** Write SQL queries for sub queries and nested queries.

#### SQL Subquery

*Subqueries* are query statements tucked inside of query statements. Like the order of operations from your high school Algebra class, order of operations also come into play when you start to embed SQL commands inside of other SQL commands (subqueries). Let's take a look at a real world example involving the *orders* table and figure out how to select only the most recent order(s) in our orders table.

USE mydatabase;

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

```
SELECT MAX(day_of_order)
```

```
FROM orders
```

**Results:**

<b>day_of_order</b>
2008-08-16 00:00:00.000

Now we can throw this query into the **WHERE** clause of another **SELECT** query and obtain the results to our little dilemma.

```
USE mydatabase;
```

```
SELECT *
```

```
FROM orders
```

```
WHERE day_of_order = (SELECT MAX(day_of_order) FROM orders)
```

id	customer	day_of_order	product	quantity
3	A+Maintenance	2008-08-16 00:00:00.000	Hanging Files	14

**SQL NESTED QUERIES**

A SQL Nested query is a SELECT query that is nested inside a SELECT, UPDATE, INSERT or DELETE SQL query.


```
SELECT model FROM product
```

```
WHERE munuf_id IN( SELECT mauf_id
```

```
FROM manufacturer WHERE manufacturer='DELL')
```

**PREAPEARED BY: Balam Singh Dafouti**



	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### Assignment Problems

1. Retrieve all the orders placed by a client named 'Rahul Desai' from the sales\_order table.

Table-name: client\_master

Client_No	Name	Bal_Due
C00001	Ashok Mehra	500
C00002	Vishal Parikh	1000
C00003	Ajay Mehta	0
C00004	Rohit Roy	0
C00005	Nalini Deewan	0
C00006	Prem Iyer	0
C00007	Rahul Desai	0

Table-name: sales\_order


Order_No	Client_No	Order_Date
019001	C00006	12-Apr-97
019002	C00002	25-Dec-97
019003	C00007	03-Oct-97
019004	C00005	18-Jun-97
019005	C00002	20-Aug-97
019006	C00007	12-Jan-97

2. Retrieve the name of all personnel who work in Mr. Pradeep's department and have worked on an inventory control system as well, from the tables emp and inv\_sys which are given below:

Table name: emp

Emp_No	Ename	Dept_No
1	Rahul	D01
2	Joshi	D02

**PREAPEARED BY: Balam Singh Dafouti**


	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

3	Lenna	D01
4	Ashwariya	D01
5	Pradeep	D01
6	Arjun	D01
7	Pritam	D01
8	Sangeeta	D02
9	Prashant	D02
10	Melba	D02

Table name: inv\_sys

Ename	Performance
Rahul	Good
Joshi	Average
Lenna	Excellent
Pradeep	Excellent
Pritam	Ok
Sangeeta	Excellent
Melba	Good

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

\*\*\*\*\*

## **LAB EXPERIMENT NO:8**

**OBJECTIVE:** Write program by using PL/SQL.

**PL/SQL (Procedural Language/Structured Query Language)** is Oracle Corporation's proprietary procedural extension to the SQL database language, used in the Oracle database. Some other SQL database management systems offer similar extensions to the SQL language.

### **Functions**

Functions in PL/SQL are a collection of SQL and PL/SQL statements that perform a task and should return a value to the calling environment.

```
CREATE OR REPLACE FUNCTION <function_name>
```

```
IS/AS
```

```
{ Variable declaration }
```


```
{ CONSTANT declaration }
```

```
RETURN return_type
```

```
BEGIN
```

```
PL/SQL Block;
```

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

## EXCEPTION

EXCEPTION Block;

END;

## Numeric variables

variable\_name number(P[,S]) := value;

To define a numeric variable, the programmer appends the variable type **NUMBER** to the name definition. To specify the (optional) precision(P) and the (optional) scale (S), one can further append these in round brackets, separated by a comma. ("Precision" in this context refers to the number of digits which the variable can hold, "scale" refers to the number of digits which can follow the decimal point.)

A selection of other datatypes for numeric variables would include:

binary\_float, binary\_double, dec, decimal, double precision, float, integer, int, numeric, real, smallint, binary\_integer

## Character variables


variable\_name varchar2(L) := 'Text';

To define a character variable, the programmer normally appends the variable type **VARCHAR2** to the name definition. There follows in brackets the maximum number of characters which the variable can store.

Other datatypes for character variables include:

varchar, char, long, raw, long raw, nchar, nchar2, clob, blob, bfile

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

Datatypes for specific columns

DECLARE

```
TYPE t_address IS RECORD (
    name address.name%TYPE,
    street address.street%TYPE,
    street_number address.street_number%TYPE,
    postcode address.postcode%TYPE);
```

```
v_address t_address;
```

BEGIN

```
SELECT name, street, street_number, postcode INTO v_address FROM address
WHERE ROWNUM = 1;
END;
```

**Assignment Problems:**

(1) Write a PL/SQL code block that will accept an account number from the user and debit an amount of Rs. 2000 from the account if the account has a minimum balance of 500 after the amount is debited. The process is to be fired on Accounts table


Account_Id	Name	Bal
AC001	Anuj	5000
AC002	Robert	10000
AC003	Mita	5000
AC004	Sunita	150000
AC005	Melba	100000

(2) Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius & the corresponding values of calculated area in a table, Areas.

(3) Write a PL/SQL block of code for inverting a number 5639 to 9365.

\*\*\*\*\*

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

## LAB EXPERIMENT NO :9

**OBJECTIVES:** Concepts of ROLL BACK, COMMIT and CHECK POINTS.

Rollback:

In SQL, ROLLBACK is a command that causes all data changes since the last BEGIN WORK, or START TRANSACTION to be discarded by the relational database management system (RDBMS), so that the state of the data is "rolled back" to the way it was before those changes were made.

In most SQL dialects, ROLLBACKs are connection specific. This means that if two connections are made to the same database, a ROLLBACK made in one connection will not affect any other connections. This is vital for proper concurrency.

### **ROLLBACK Statement**

The ROLLBACK Statement terminates the current transaction and rescinds all changes made under the transaction. It *rolls back* the changes to the database. The ROLLBACK statement has the following general format:

**ROLLBACK [WORK]**

WORK is an optional keyword that does not change the semantics of ROLLBACK


### **COMMIT Statement**

The COMMIT Statement terminates the current transaction and makes all changes under the transaction persistent. It *commits* the changes to the database. The COMMIT statement has the following general format:

**COMMIT [WORK]**

WORK is an optional keyword that does not change the semantics of COMMIT.

**PREPARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

### **CHECKPOINT**

Forces all dirty pages for the current database to be written to disk. Dirty pages are data or log pages that have been modified after they were read into the buffer cache, but the modifications have not yet been written to disk. If the current database is in log truncate mode, CHECKPOINT also truncates the inactive portion of the log. For more information about log truncation, Truncating the Transaction Log

### *Syntax*

CHECKPOINT

### **Assignment Problem:**

(1) Write a PL/SQL block of code that first inserts a record in an 'Emp' table. Update the salaries of Blake and Clark by Rs. 2000 and Rs. 1500. Then check to see that the total salary does not exceeded 20000. If the total salary is grater than 20000 then undo the updates made to the salaries of Blake and Clark. Table ' Emp' is given below:

Emp_No	Emp_Name	Sal
E001	Harry	5000
E002	Blake	1000
E003	Jack	5000
E004	Clark	1000


### **LAB EXPERIMENT NO:10**

**OBJECTIVE :** Create Views, Cursors and triggers and write assertions.

#### **SQL CREATE VIEW Statement**

In SQL, a view is a virtual table based on the result-set of an SQL statement.

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

### SQL CREATE VIEW Syntax

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

### CREATE CURSOR

In SQL procedures, a cursor make it possible to define a result set (a set of data rows) and perform complex logic on a row by row basis. By using the same mechanics, an SQL procedure can also define a result set and return it directly to the caller of the SQL procedure or to a client application.


A cursor can be viewed as a pointer to one row in a set of rows. The cursor can only reference one row at a time, but can move to other rows of the result set as needed.

```
CREATE CURSOR alias_name
(fname1 type [(precision [, scale])] [NULL | NOT NULL]
[CHECK lExpression [ERROR cMessageText]]
[AUTOINC [NEXTVALUE NextValue [STEP StepValue]]]
[DEFAULT eExpression] [UNIQUE [COLLATE cCollateSequence]]
[NOCPTRANS] [, fname2 ...])
| FROM ARRAY ArrayName
```

### CREATE TRIGGER

**PREAPEARED BY: Balam Singh Dafouti**



	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

Creates a trigger, which is a special kind of stored procedure that executes automatically when a user attempts the specified data-modification statement on the specified table. Microsoft® SQL Server™ allows the creation of multiple triggers for any given INSERT, UPDATE, or DELETE statement.

### **Syntax**


```

CREATE TRIGGER trigger_name
ON { table | view }
[ WITH ENCRYPTION ]
{
    { { FOR | AFTER | INSTEAD OF } { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
    [ WITH APPEND ]
    [ NOT FOR REPLICATION ]
    AS
    [ { IF UPDATE ( column )
      [ { AND | OR } UPDATE ( column ) ]
      [ ...n ]
    | IF ( COLUMNS_UPDATED ( ) { bitwise_operator } updated_bitmask )
      { comparison_operator } column_bitmask [ ...n ]
    } ]
    sql_statement [ ...n ]
}
}

```

### **Assignment Problems:**

<b>PREAPEARED BY: Balam Singh Dafouti</b>
---

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

(1) Write a PL/SQL code block that determines the top three highest paid employees from the employee table. Use appropriate cursor attributes for the same. Display the name and salary of these employees. Table 'Emp' is given below:


Emp_No	Emp_Name	Salary
E001	Mita	10000
E002	Vaishali	1000
E003	Lakshmi	3500
E004	Praveen	4000
E005	Aditya	2500
E006	Sunita	2500
E007	Ajay	2000
E008	Melba	2100

(2) Create View on the above table where employee salary is more than 3500.

(3) Create a transparent audit system for a table Client\_master. The system must keep track of the records that are being deleted or updated. The functionality being when a record is deleted or modified the original record details and the date of operation are stored in the deleted or update is allowed to go through.

Product no	Description	Profit percent	Unit_measure	Qty on hand	Recorder level	Sell price	Cost price
P00001	1.44 Floppies	5	Piece	100	20	525	500
P03453	Monitor	6	Piece	10	3	12000	11280
P06734	Mouse	5	Piece	20	5	1050	1000
P07865	1.22 Floppies	5	Piece	100	20	525	500
P07868	Keyboards	2	Piece	10	3	3150	3050
P07885	CD Drive	2.5	Piece	10	3	5250	5100
P07965	540 HDD	4	Piece	10	3	8400	8000
P07975	1.44 Drive	5	Piece	10	3	1050	1000
P08865	1.22 Drive	5	Piece	2	3	1050	1000

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

## LAB EXPERIMENT NO. :11

**OBJECTIVE:** Create FORMS and REPORTS

### Creating a Form

Sometimes it is useful to collect data from your website users and store this information in a SQL database. We have already seen you can populate a database using PHP, now we will add the practicality of allowing the data to be added through a user friendly web form.

The first thing we will do is create a page with a form. For our demonstration we will make a very simple one:


```

<form action="process.php" method="post">
Your Name: <input type="text" name="name"><br>
E-mail: <input type="text" name = "email"><br>
Location: <input type="text" name = "location"><br>
<input type="submit" value="Submit">
</form>

```

### Insert Into - Adding Data from a Form

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

Next you need to make process.php, the page that our form sends its data to. Here is an example of how to collect this data to post to the SQL database:

```
<?
$name=$_POST['name'];
$email=$_POST['email'];
$location=$_POST['location'];
mysql_connect("your.hostaddress.com", "username", "password") or die(mysql_error());
mysql_select_db("Database_Name") or die(mysql_error());
mysql_query("INSERT INTO `data` VALUES ('$name', '$email', '$location')");
Print "Your information has been successfully added to the database.";
?>
```

### **Add File Uploads**

Now you know how to store user data in MySQL, so let's take it one step farther and learn how to upload a file for storage. First let's make our sample database:


```
CREATE TABLE uploads (id INT(4) NOT NULL AUTO_INCREMENT PRIMARY
KEY, description CHAR(50), data LONGBLOB, filename CHAR(50), filesize
CHAR(50), filetype CHAR(50) );
```

Next, we will create a form to allow the user to upload her file. This is just a simple form, obviously you could dress it up if you wanted:

```
<form method="post" action="upload.php" enctype="multipart/form-data">
Description:<br>
<input type="text" name="form_description" size="40">
<input type="hidden" name="MAX_FILE_SIZE" value="1000000">
<br>File to upload:<br>
<input type="file" name="form_data" size="40">
<p><input type="submit" name="submit" value="submit">
</form>
```

**Adding File Uploads to SQL** Next we need to actually create upload.php, which will take our users file and store it in our database. Below is sample coding for upload.php.

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University, Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

```
<?php
mysql_connect("your.server.com","username","password");
mysql_select_db("database_name");
$data = addslashes(fread(fopen($form_data, "r"), filesize($form_data)));
$result=MYSQL_QUERY("INSERT INTO uploads (description,
data,filename,filesize,filetype) ". "VALUES
```

### **Retrieving Files**

```
<?php
mysql_connect("your.server.com","username","password");
mysql_select_db("database_name");
$query = "SELECT data,filetype FROM uploads where id=$id";
$result = MYSQL_QUERY($query);
$data = MYSQL_RESULT($result,0,"data");
$type = MYSQL_RESULT($result,0,"filetype");
Header( "Content-type: $type");
print $data;
?>
```


Now to retrieve our file, we point our browser to:  
<http://www.yoursite.com/download.php?id=2> (replace the 2 with whatever file ID you want to download/display)

### **Removing Files**

```
<?php
mysql_connect

("your.server.com","username","password");
mysql_select_db("database_name");
$query = "DELETE FROM uploads where id=$id";
$delete = MYSQL_QUERY($query);
```

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

```
print "File ID $id has been removed from the database";
?>
```

### CREATING REPORTS :

#### **Elements of a Report**


- **Queries** define the data to be retrieved from the database.
- **Fields** represent column expressions and report calculations from SELECT statements and describe how each is to be displayed.
- **Groups** contain sets of fields. Groups are used to describe each section or subsection in the report and its relationships, as well as to dictate control breaks for subtotalling purposes.
- **Summaries** are similar to fields, except they display subtotals and grandtotals rather than data directly from a query.
- **Texts** contain fields, summaries, and parameter references, combined with literal strings, such as titles, and define the final report format.
- The **Report** object defines the page size, margins, parameter form text, comments, security, and history of the report.
- **Parameters** contain literal values that you supply at runtime to control the behavior of the report. You can vary the data, routing of the output, and select national language settings.

#### Report Building Process

There are a few simple steps involved in building a report:

- creating a blank report
- specifying the data and calculations to be used in the report

**PREAPEARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University,</b> <b>Haldwani</b> <b>School of Computer Science &amp; IT</b>	<b>LABORATORY</b> <b>MANUAL</b>
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.:Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

- specifying the format of the report

SQL\*ReportWriter separates the process of data access from report formatting, providing you with an additional degree of freedom

### Screen Layout

The Main Menu appears when SQL\*ReportWriter is invoked, when [Accept] is pressed from any setting screen or menu, or when [Menu] is pressed.

The status line displays the following information:

- the name of the current report when one is open
- whether or not the current field has a List of values obtained by pressing [List].
- the status of insert/replace toggle for editing purposes
- the status of the Query mode toggle and the highlight indicator

### Action Choice:

The Action Menu contains a list of options which operate on report definitions, including: New, Open, Copy, Rename, Drop, Execute, Generate and Quit.


**NEW:** Creates a new report definition. The default margins are set on the Report Screen, default Report and Page text objects are created (blank) on the Text Screen, and system parameters for DESTYPE, DESNAME, DESFORMAT, COPIES, CURRENCY, THOUSANDS, and DECIMAL are automatically created on the parameter screen.

**OPEN:** Opens an existing report definition for editing. You must enter the name of the report to open by either typing it in or choosing it from List of values, which displays the names of your own reports.

**COPY:** Creates a new report by copying an existing report definition.

**RENAME:** Changes the name of an existing report definition.

**PREPARED BY: Balam Singh Dafouti**

	<b>Uttarakhand Open University, Haldwani</b> School of Computer Science & IT	LABORATORY MANUAL
	<b>PRACTICAL INSTRUCTION SHEET</b>	
DEPTT.: Computer Science & IT	LABORATORY: Database Management System (BCA-09)	SEMESTER: III

**DROP:** Deletes the specified report definition.

**EXECUTE:** Generates and runs an existing report.

**GENERATE:** Creates a runfile for an existing report.

**QUIT:** Exits to the operating system.

### **Assignment Problems:**

(1) Create a form which has following fields:

(i) id

(ii) name

(iii) class

(iv) section

(2) Enter the some values into the form and upload these values into the table 'Student'.

(3) Retrieve all the values from the table and display.

(4) Create a report for the above form.

**PREAPEARED BY: Balam Singh Dafouti**