



Practical no: 1

Aim: Design Schema for College Management System

Theory:

* Relational Model concept :

- Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has name or attribute.
- 1) Domain : It contains a set of atomic values that an attribute can take.
- 2) Attribute : It contains the name of a column in a particular table. Each attribute A_i must have a domain, $\text{dom}(A_i)$.
- 3) Relational instance : In the relational database system, the relational instance is represented by a finite set of tuples. Relation instance do not have duplicate tuples.
- 4) Relational schema : A relational schema contains the name of the relation and name of all columns or attributes.
- 5) Relational Key : In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.



example : Student Relation (Table)

Sid	Sname	Address	Branch
001	Kunal	Pune	CSE
002	Ram	Thane	ETC
003	Ajay	Pune	civil

- In the above student relation

- 1) Attributes

- Sid, Sname, Address, Branch

- 2) Relation Schema

- $\text{Student}(\text{Sid}, \text{Sname}, \text{Address}, \text{Branch})$

- Tuple : each row in relation is called as tuple

- 3) Degree (Number of Attributes)

- 4

- 4) Cardinality (Number of Tuples)

- 3

- 5) Domain constraints

- a) Sid : int

- b) Sname : Varchar

- c) Address : Varchar

- d) Branch : Varchar

- 6) Relational key / Key constraint : Sid is key in student Relation



* Integrity Constraints :

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating and other processes have to be performed in such a way that data integrity is not affected.

• Types of Integrity Constraint

i) Domain constraints:

- It can be defined as the definition of valid set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency. The value of the attribute must be available in the corresponding domain.

ii) Entity integrity constraint

- The entity integrity constraint states that primary key value can't be null. This is because the primary key value is used to identify individual row in relation and if the primary key has a null value that we can't identify those rows.
- A table can contain a null value other than the primary key field.

iii) Referential integrity constraint

- A referential integrity constraint is specified between two tables.



- Referential integrity constraint in DBMS are based on concept of foreign key when one attribute of the relation can only fetch value from other attributes of same relation or any other relation is called Referential integrity constraints.

e.g :-

sid	sname	city	b-No	→ Foreign key
1	Ajay	Ahad	1	
2	Ram	Pune	2	
3	Ravi	Dhule	1	
4	Kunal	Pune	3	



b-No	bname
1	CSE
2	ETC
3	civil

4) Key constraints:

- Keys are the entity set that is used to identify an entity within its entity set uniquely
- An entity set can have multiple keys, but out of which one key will be the primary key.

e.g: In. Student Relation : Roll.no, Mobile.no, Aadhar No, Registration no. are the multiple keys.



Practical No:2

Aim: Creating tables, Renaming tables, Data constraints (Primary Key, Foreign key, Not Null), Data insertion into a table.

Theory :

* SQL commands (DDL /DML) :

1) CREATE :

Syntax : Create database name_database

- The create command is used to create a new SQL database.

e.g: create database college ;

2) ALTER :

Syntax : - Alter table command is used to add modify or drop the columns in existing table .

a) Add

- Alter table.name add (column1 datatype ,.... columnn datatype)

e.g: Alter table student add (deptno int)

b) Modify

- Alter table.name modify (column1 datatype(size))

e.g: Alter table student Modify city varchar(30)



c) drop

- Alter table name drop (column 1 ...) ;
e.g: Alter table student drop Address ;

3) TRUNCATE :

Syntax : Truncate Table table-name ;

- The Truncate command deletes the data inside a table, but not the table itself
e.g: Truncate table student ;

4) DROP :

Syntax: Drop Table table-name ;

- The Drop command is used to drop an existing table in a database .

5) RENAME :

Syntax: Rename table-name to new-table-name ;

- The Rename Table command is used to change the table name .
- We can also use the Alter Table statement to change the table name .

6) INSERT :

- . Insert (or) Insert Into statement is used to insert new records in a table .

Syntax : INSERT INTO table-name (column1, ...) Values (value1, ...)



Chhatrapati Shahu Maharaj Shikshan Sanstha's
CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :

e.g.: INSERT INTO Student (Sid , Sname ,
Branch , Address)
values ('3130' , "Ram" , 'CSE' , 'Aurangabad')

7) UPDATE :

Syntax - Update table-name
SET column 1 = value1 , . . .
where condition

- The UPDATE command/statement is used to modify the existing records in a table .

e.g.: UPDATE Student
SET Sname = 'Kunal' , Address = 'Pune'
where Sid = '3130' ;

8) DELETE :

Syntax : The Delete From table-name
where condition ;

- The DELETE statement is used to delete existing records in a table .

e.g.: DELETE from student
where Sname = 'Ram' ;



Practical No. 3

Aim: Implementing SQL commands for Grouping and Aggregate functions.

Theory :

* SQL clause :

1) Group By :

- The Group By statement groups rows that have the same values into summary rows, like "find the number of customers in each country".
- GroupBy statement is use with aggregate function .
i.e.: COUNT(), MAX(), MIN(), SUM(), AVG() .
- Syntax:

Select column(s)

Aggregate function (column-name)
from tablename

GroupBy columnname

2) Order By :

- The OrderBy keyword is used to sort the result-set in ascending or descending order.
- The OrderBy keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.



- Syntax : Select column(s)
from tablename
Order by columnname (ASC/DESC)

* SQL Aggregate Functions:

1) COUNT : This function returns the number of rows that matches a specific criteria.

- Syntax : Select count (column-name)
from table-name
where condition

e.g : Select count (sid) from student

2) AVG : It return the average value of specified column.

- Syntax : Select AVG (column-name)
from table-name
where condition

e.g : Select AVG (marks)
from student

3) SUM :

This function returns sum of specified column.

- Syntax : Select SUM (column-name)
from table-name
where condition

e.g : Select sum (marks)
from student



4) Min :-

- The Min () return the minimum value of a set var

- Syntax : Select min (^{columnname} marks)
from tablename
where condition

e.g:- Select min (marks)
from student

5) Max :-

- The Max () return the maximum value of a set

- Syntax : Select max (columnname)
from tablename
where condition -

e.g:- Select max (marks)
from student



Practical no: 4

Aim: Implementing SQL commands for subqueries,
Set operations, Joins

Theory:

* SQL Operators:

Operators are the foundation of any programming language. We can define operators as symbols that help us to perform specific mathematical and logical computations on operands.

- Logical operators: The logical operators are those that are true or false. They return true or false values to combine one or more true or false values.

1) AND: AND compares between two booleans as expressions and returns true when both expressions are true.

Syntax:

Select * From table-name

where condition1 AND condition2 AND
.... conditionⁿ;

2) OR: OR compares between two booleans as expressions and returns true when one of the expressions is true.



Syntax:

Select * from table-name;

Where condition1 OR condition2 OR ... conditionn;

3) NOT : It takes a single boolean value as argument and changes its value from false to true or from true to false.

Syntax: Select column1, column2, ...

From table-name;

Where Not condition;

- Some special operators:

1) ALL : ALL is used to select all records of a select statement. It compares a value to every value in a list of results from a query.

The ALL must be preceded by the comparison operators and evaluates to true if the query returns no rows.

Syntax: Select ALL yield_name

From table-name

Where condition(s);

2) ANY : ANY compares a value of the first table with all values of the second table and returns the row if there is a match with any value.



Syntax : Select column-name
From table-name
Where test-expression comparison
ANY () ;

3) Between : It is used with where clause
to match values in a range

Syntax : Select column-name(s)
From Table-name
Where column-name BETWEEN
Range 1 AND Range 2 ;

4) EXISTS : Exists operator executes the
outer SQL query if the subquery is not
Null .

Syntax :
Select column-name(s)
From table-name
Where EXISTS (
Select column-name(s)
From table-name
Where condition) ;



* SQL Join :

- The SQL join joins two tables based on a common column, and selects records that have matching values in the columns.

1) Inner Join : The inner join selects records that have matching value in both the dates.

Syntax : Select column(s)

from Table1 InnerJoin Table2

ON Table1.colⁿ = Table2.colⁿ

e.g : Select s.sid , s.sname , d.dname
From student as s Inner Join
department as d
ON s.deptid = d.deptid .

2) Left Join : It returns all the records from the left table and matching records from the right table.

Syntax : Select column(s)

From Table1 LeftJoin Table2

ON Table1.colⁿ = Table2.colⁿ

e.g : Select s.sid , s.sname , d.dname
From student as s Left Join
department as d
ON s.deptid = d.deptid .

ON s.deptid = d.deptid .



3) Right Join : It returns all the records from the right table and matching records from the left table.

- Syntax : Select column(s)

from Table1 Right Join Table2
on Table2.colⁿ = Table2.colⁿ;

e.g : Select s.sid , s.sname , d.dname
from student as s Right Join
department as d
on s.dept id = d.dept id ;

4) Full outer Join : It joins two tables based on a common column, and selects records that have matching values in these columns and remaining rows from both of the tables.

- Syntax : Select column(s)

From Table1 full outer Join Table2
on Table1.colⁿ = Table2.colⁿ;

e.g : Select s.sid , s.sname , d.dname
from student as s full outer join
department as d
on s.dept id = d.dept id ;



* SQL Set Operations -

1) UNION : Union operation is used to combine the result of two more SQL queries. In union operation the number of columns in both table must be same.

Syntax : Select column(s) from Table 1
UNION
Select column(s) from Table 2

e.g.: Select * from A
UNION
Select * from B

2) UNION ALL : It is equal to the Union operation. It returns the set without removing duplication & sorting the data.

Syntax : Select column-name from table 1
UNION ALL

Select column-name from table 2;

e.g.: Select * from A
UNION ALL
Select * from B

3) INTERSECT : It is used to combine two SELECT statement. The intersect operation returns the common rows from both the select statements.

Syntax : Select column(s) from Table 1
Intersect
Select column(s) from Table 2