DATABASE MANAGEMENT SYSTEM

(IT-401)

LAB MANUAL

B.TECH- INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

UNIVERSITY INSTITUTE OF TECHNOLOGY

RGPV BHOPAL

DEPARTMENT OF INFORMATION TECHNOLOGY

UNIVERSITY INSTITUTE OF TECHNOLOGY

RGPV BHOPAL

Vision of Department

To transform young minds into competent Information Technology professionals by synergizing ethical values and adapting the technological advancements to meet global challenges.

Mission of Department

- M1 : To continuously strive for the overall development of students by educating them in latest IT technologies.
- M2 : To provide conducive environment for exploring and applying knowledge to improve interpersonal and technical skills.
- M3 : To inculcate ethics, environmental awareness and societal commitment which will contribute to personal and professional growth in the interest of society.

Program Educational Objectives

PEO1 Attain professional competency in the area of information technology.

PEO2 Apply knowledge of basic sciences and engineering to pursue higher education and research to work in multidisciplinary environment.

PEO3 Excel as a socially committed IT engineer having entrepreneurial skill, good leadership qualities and high ethical values.

PROGRAM OUTCOMES [POs]

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem Analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet t h e specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern Tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

DEPARTMENT OF INFORMATION TECHNOLOGY

UNIVERSITY INSTITUTE OF TECHNOLOGY

RGPV BHOPAL

Program Specific Outcomes

PSO1 Graduate will be able to acquire knowledge of information technology to develop efficient algorithmic solutions for engineering applications.

PSO2 Graduate will be able to utilize the skill of data analytics, IoT, cloud computing and AI to address the technological challenges.

Introduction

A DBMS is a set of software programs that controls the system organization, storage, management, and retrieval of data in a database. DBMSs are categorized according to their data structures or types. The DBMS accepts requests for data from an application program and instructs the operating system to transfer the appropriate data. The queries and responses must be submitted and received according to a format that conforms to one or more applicable protocols. When a DBMS is used, information systems can be changed more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Database servers are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions.

Components

- **DBMS Engine** accepts logical requests from various other DBMS subsystems, converts them into physical equivalents, and actually accesses the database and data dictionary as they exist on a storage device.
- **Data Definition Subsystem** helps the user create and maintain the data dictionary and define the structure of the files in a database.
- **Data Manipulation Subsystem** helps the user to add, change, and delete information in a database and query it for valuable information. Software tools within the data manipulation subsystem are most often the primary interface between user and the information contained in a database. It allows the user to specify its logical information requirements.
- **Application Generation Subsystem** contains facilities to help users develop transaction-intensive applications. It usually requires that the user perform a detailed series of tasks to process a transaction. It facilitates easy-to-use data entry screens, programming languages, and interfaces.
- **Data Administration Subsystem** helps users manage the overall database environment by providing facilities for backup and recovery, security management, query optimization, concurrency control, and change management.

Software Required: Operating System, Sql /Oracle /Ms Access/PostgreSQL/Mysql etc.

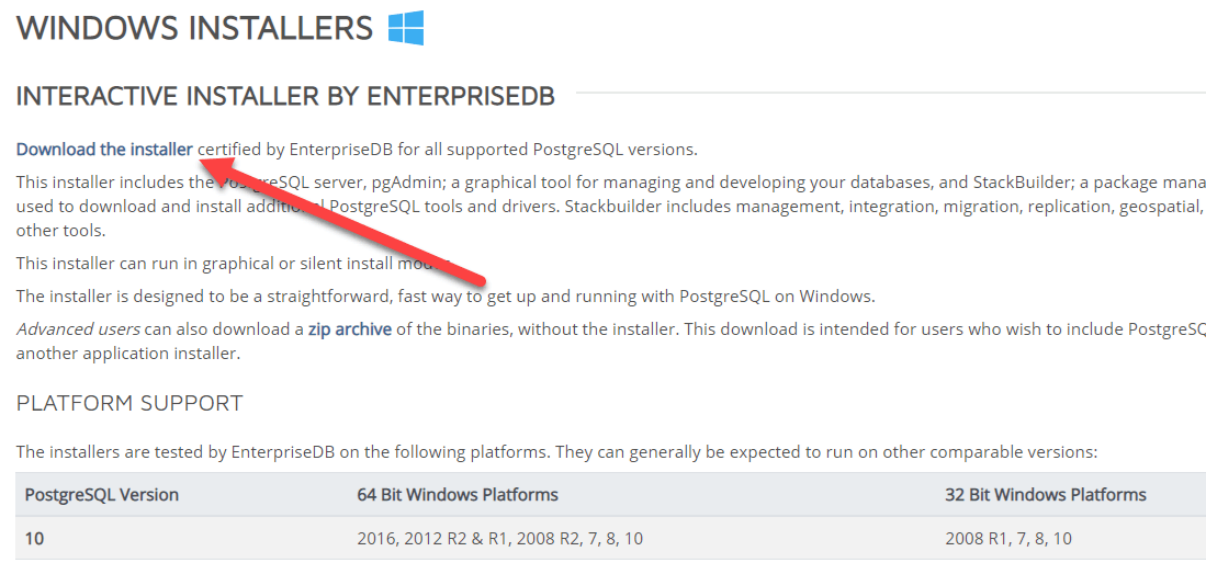
How to Download & Install PostgreSQL

Following is a step-by-step process on How to **Install PostgreSQL on Windows Machine**:

Step 1) Open your browser.
Go to https://www.postgresql.org/download and select Windows

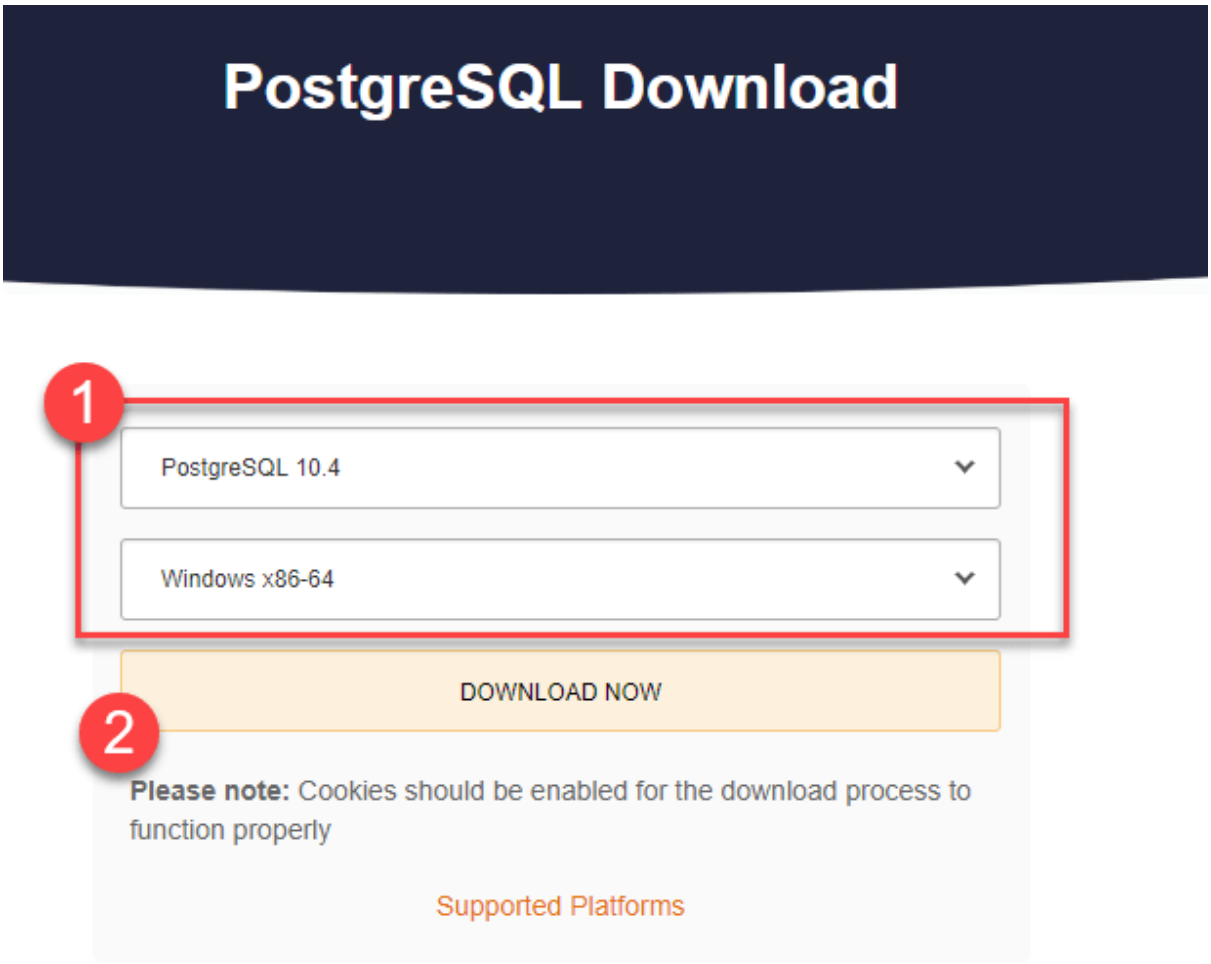
Step 2) Check options.
You are given two options 1) Interactive Installer by EnterpriseDB and 2) Graphical Installer by BigSQL.

BigSQL currently installs pgAdmin version 3 which is deprecated. It's best to choose EnterpriseDB which installs the latest version 4

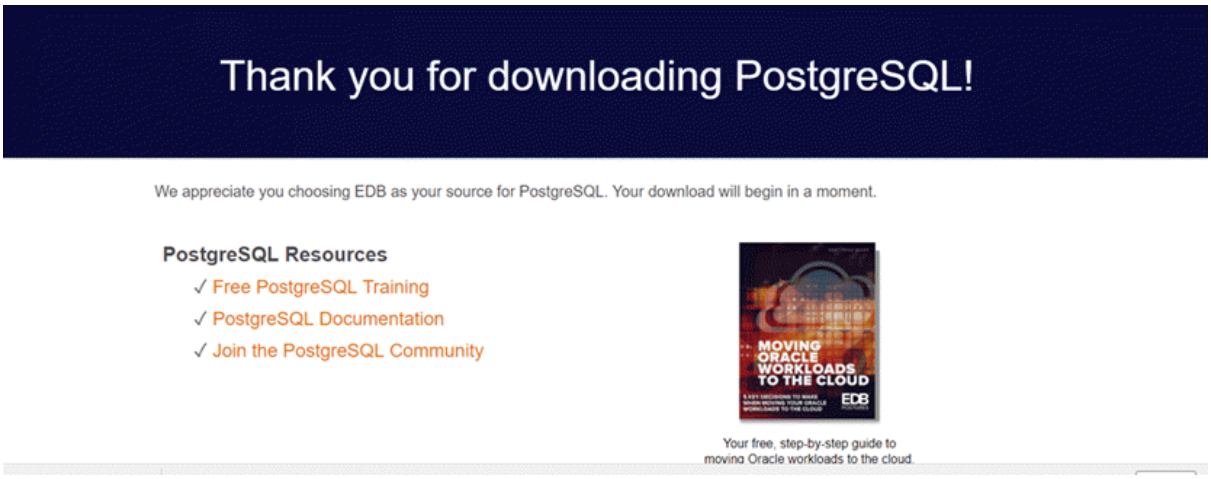


Step 3) Select PostgreSQL version.

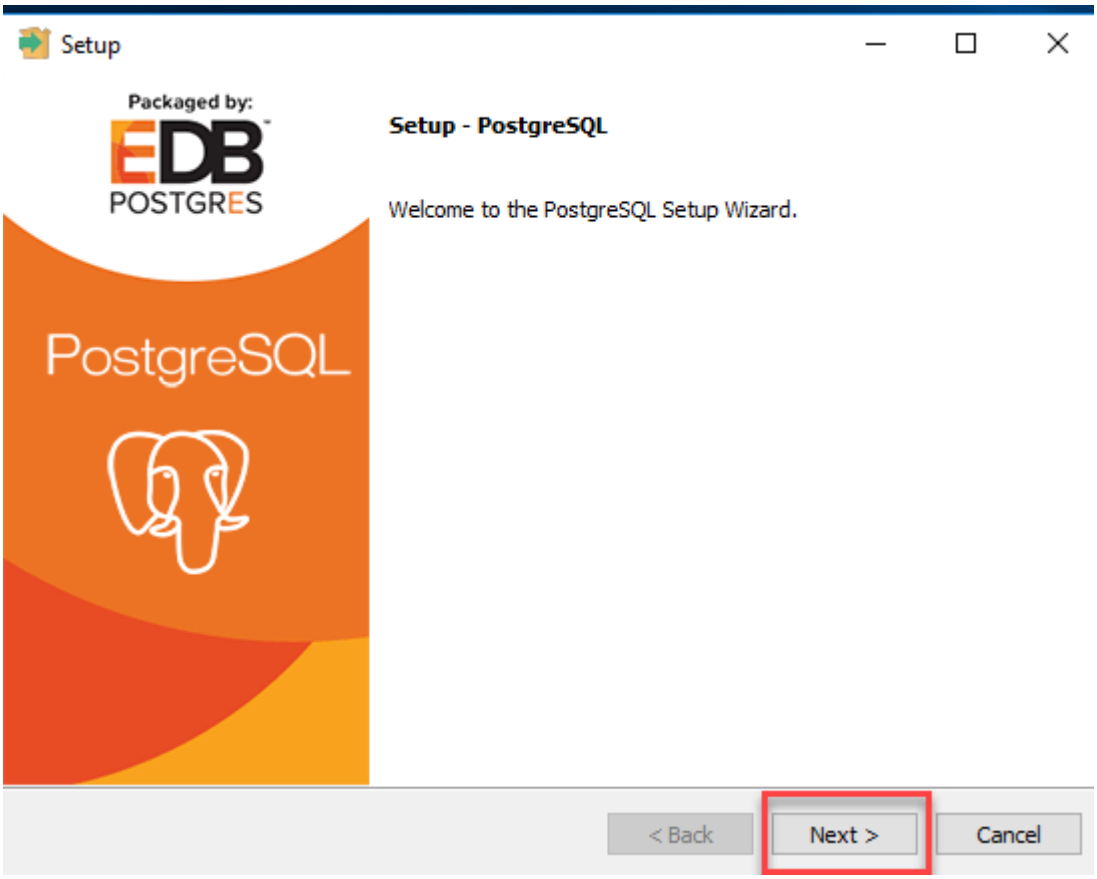
1. You will be prompted to desired PostgreSQL version and operating system. Select the latest PostgreSQL version and OS as per your environment
2. Click the Download Button



Download will begin

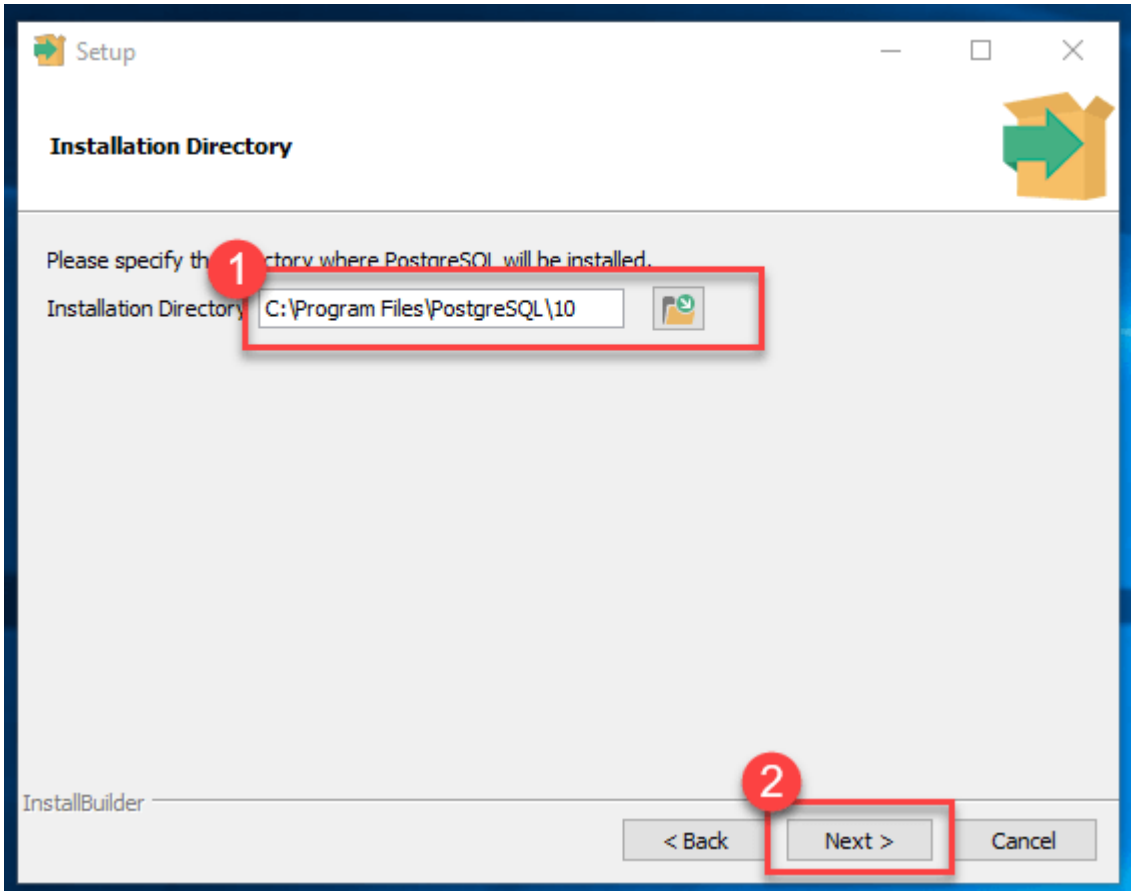


Step 4) Open exe file.
Once you Download PostgreSQL, open the downloaded exe and Click next on the install welcome screen.



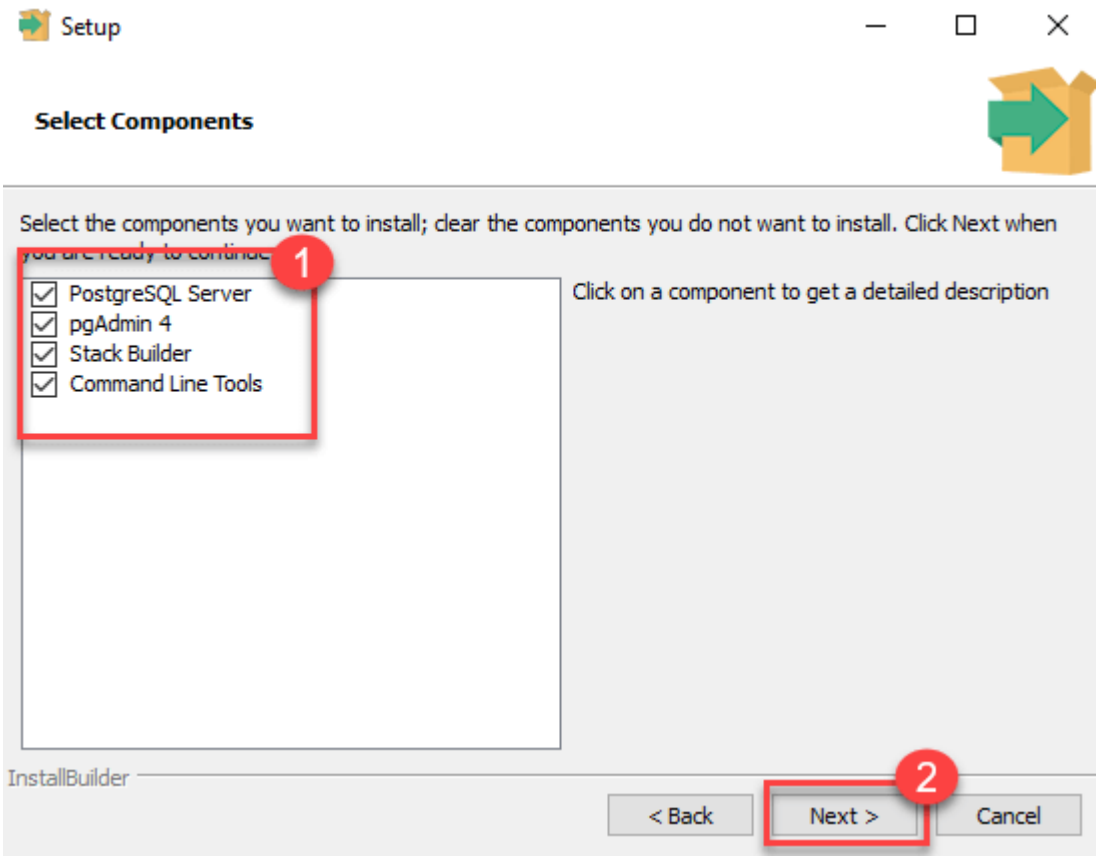
Step 5) Update location.

1. Change the Installation directory if required, else leave it to default
2. Click Next



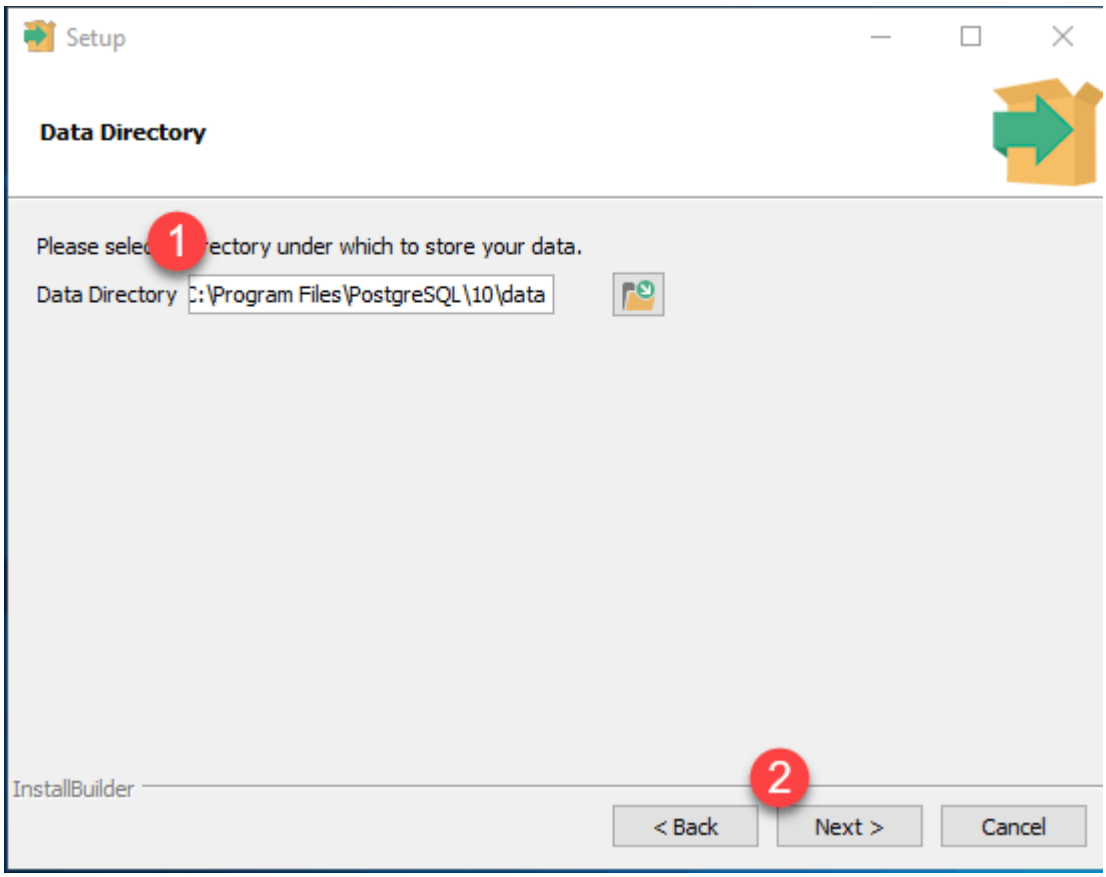
Step 6) Select components.

1. You may choose the components you want to install in your system. You may uncheck Stack Builder
2. Click Next



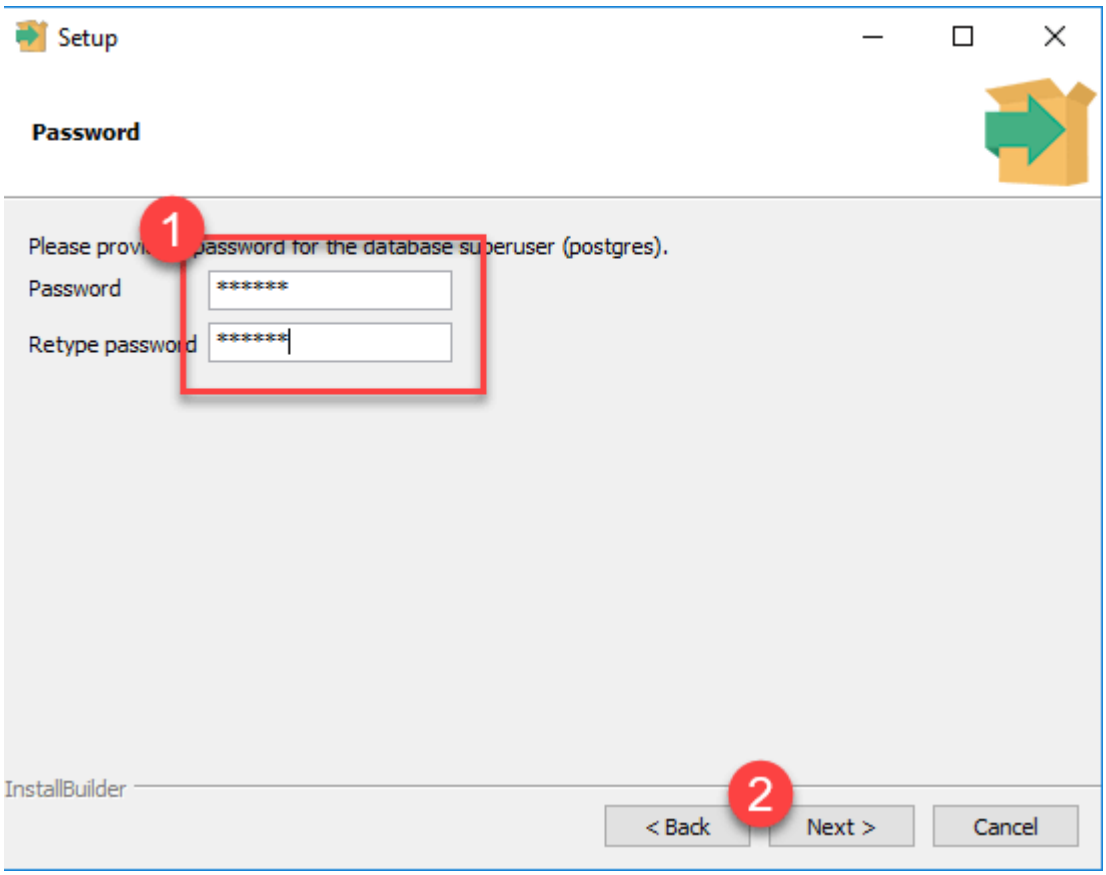
Step 7) Check data location.

1. You may change the data location
2. Click Next



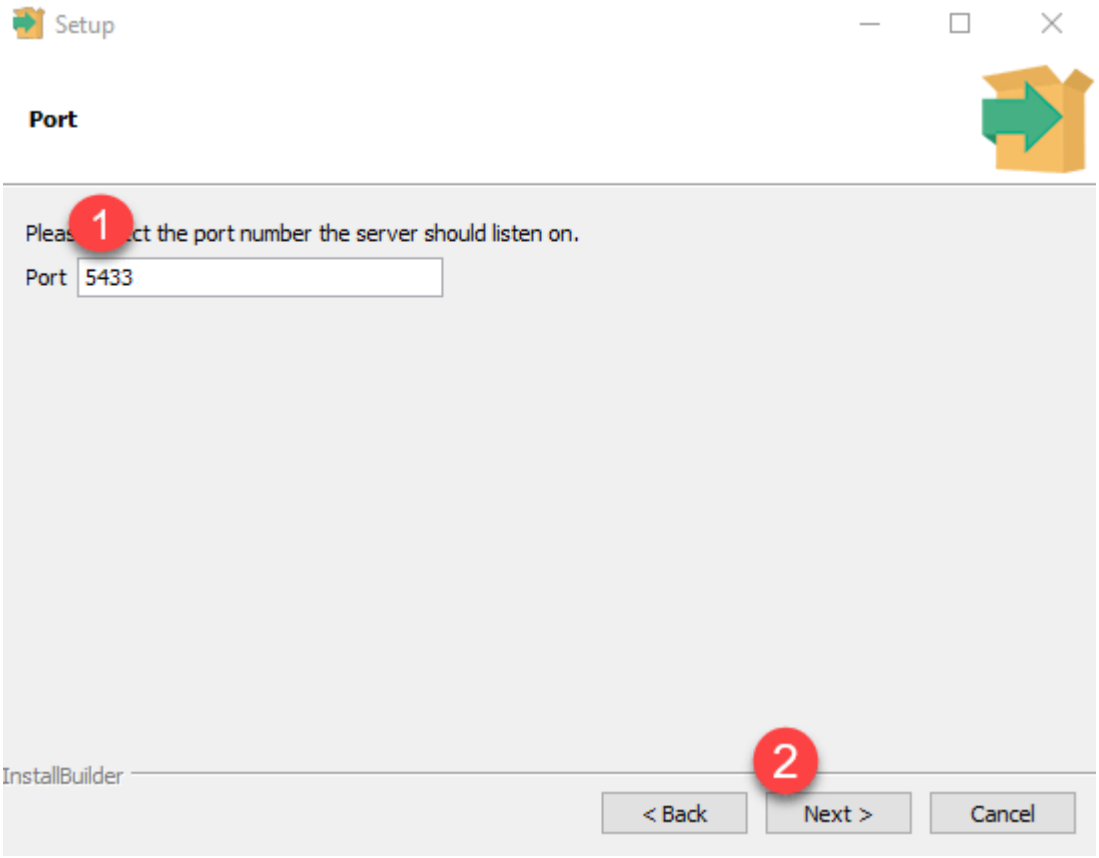
Step 8) Enter password.

1. Enter super user password. Make a note of it
2. Click Next



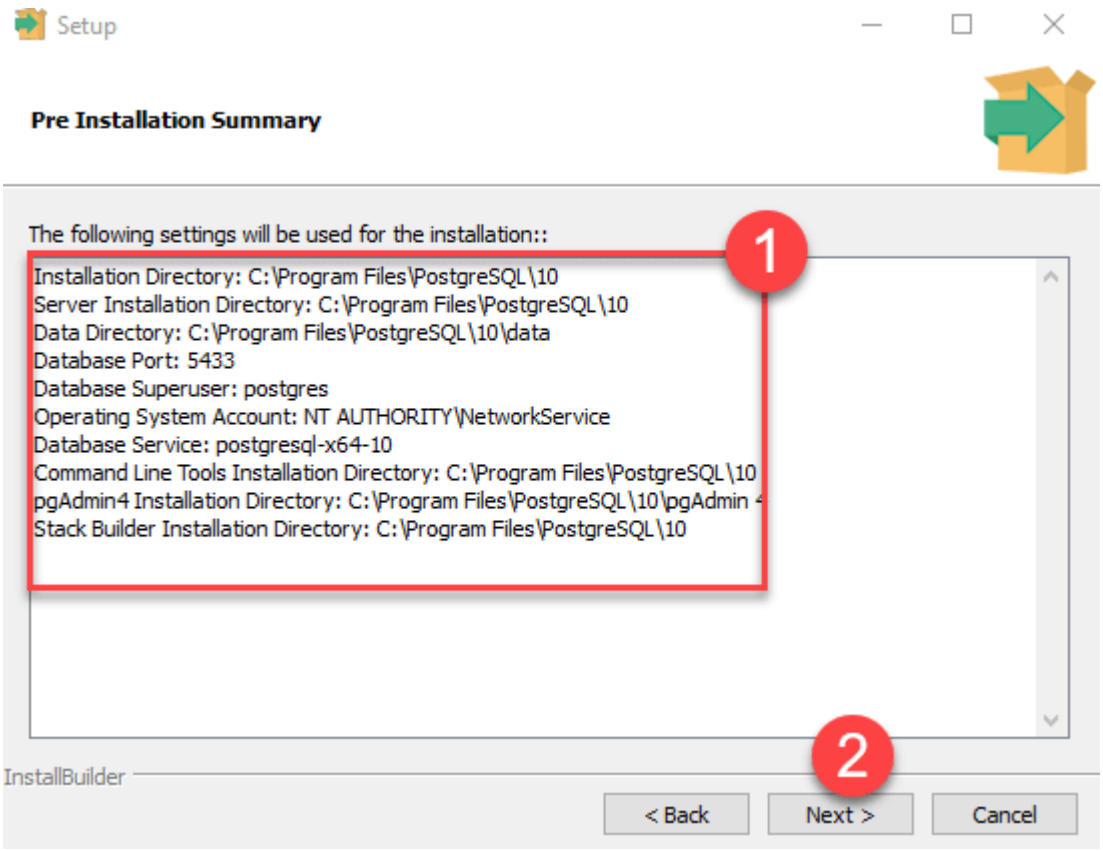
Step 9) Check port option.

1. Leave the port number default
2. Click Next

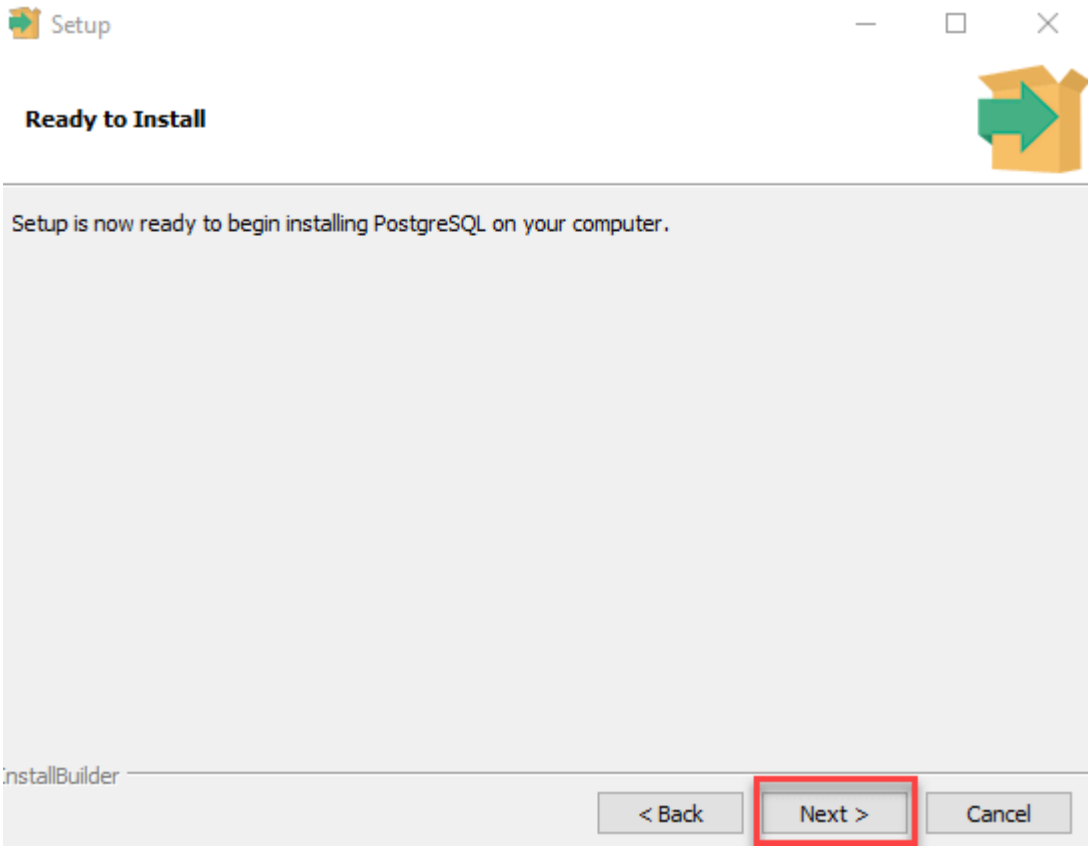


Step 10) Check summary.

1. Check the pre-installation summary:
2. Click Next



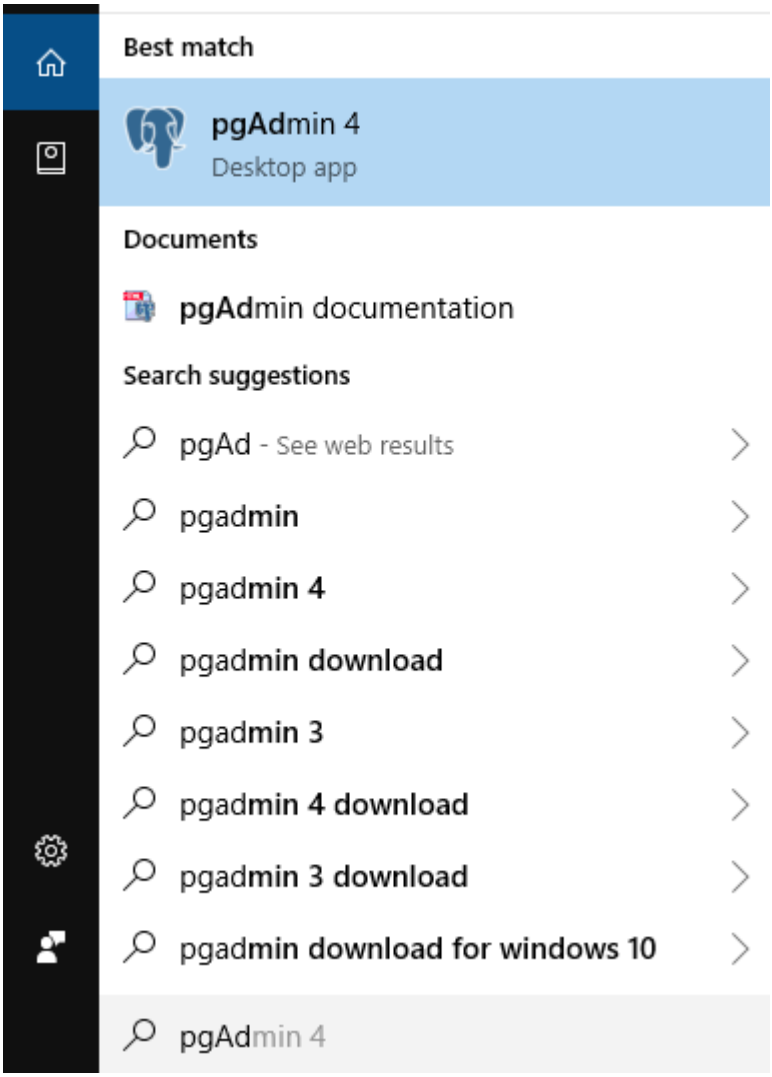
Step 11) Ready to install.
Click the next button.



Step 12) Check stack builder prompt.
Once install is complete you will see the Stack Builder prompt

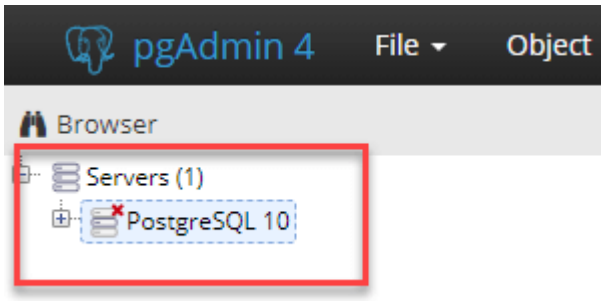
1. Uncheck that option. We will use Stack Builder in more advance tutorials
2. Click Finish

Step 13) Launch PostgreSQL.
To launch PostgreSQL go to Start Menu and search pgAdmin 4



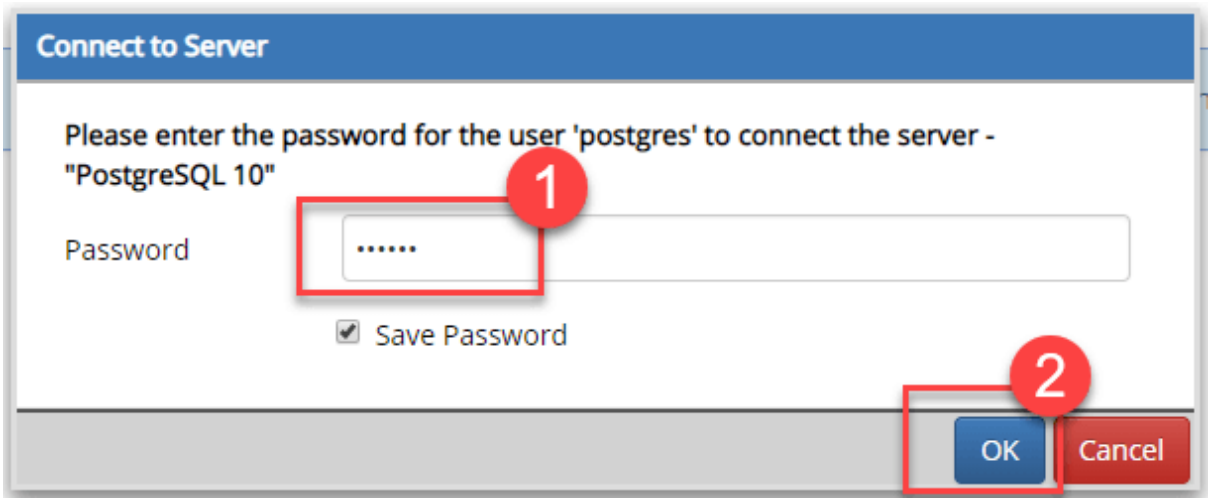
Step 14) Check pgAdmin.
You will see pgAdmin homepage.

Step 15) Find PostgreSQL 10.
Click on Servers > PostgreSQL 10 in the left tree

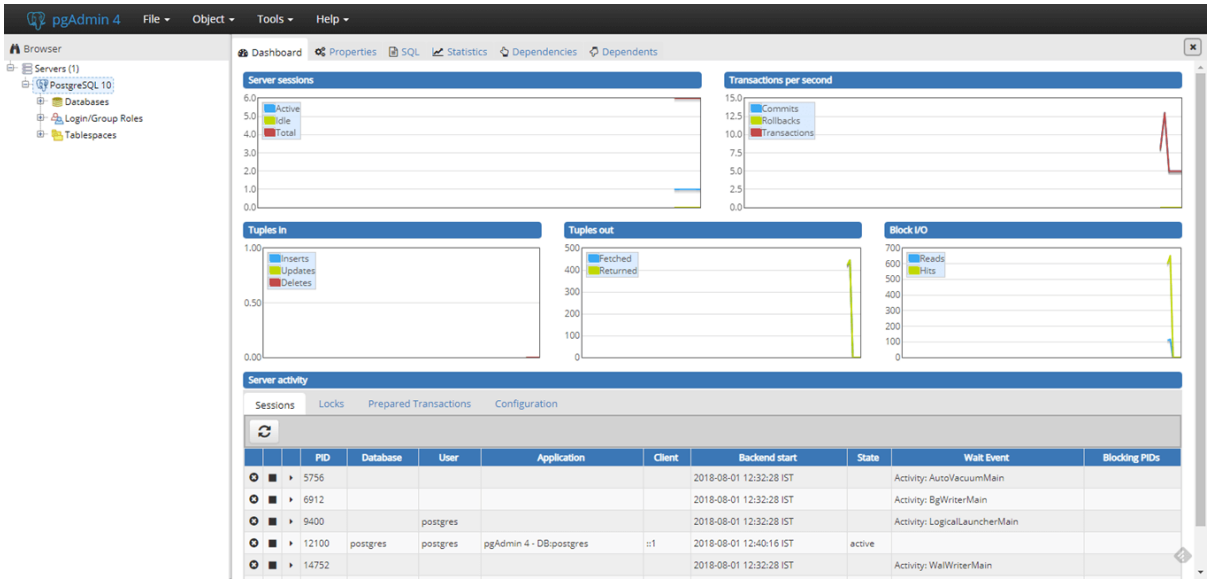


Step 16) Enter password.

1. Enter super user password set during installation
2. Click OK



Step 17) Check Dashboard.
You will see the Dashboard



That's it to PostgreSQL installation.

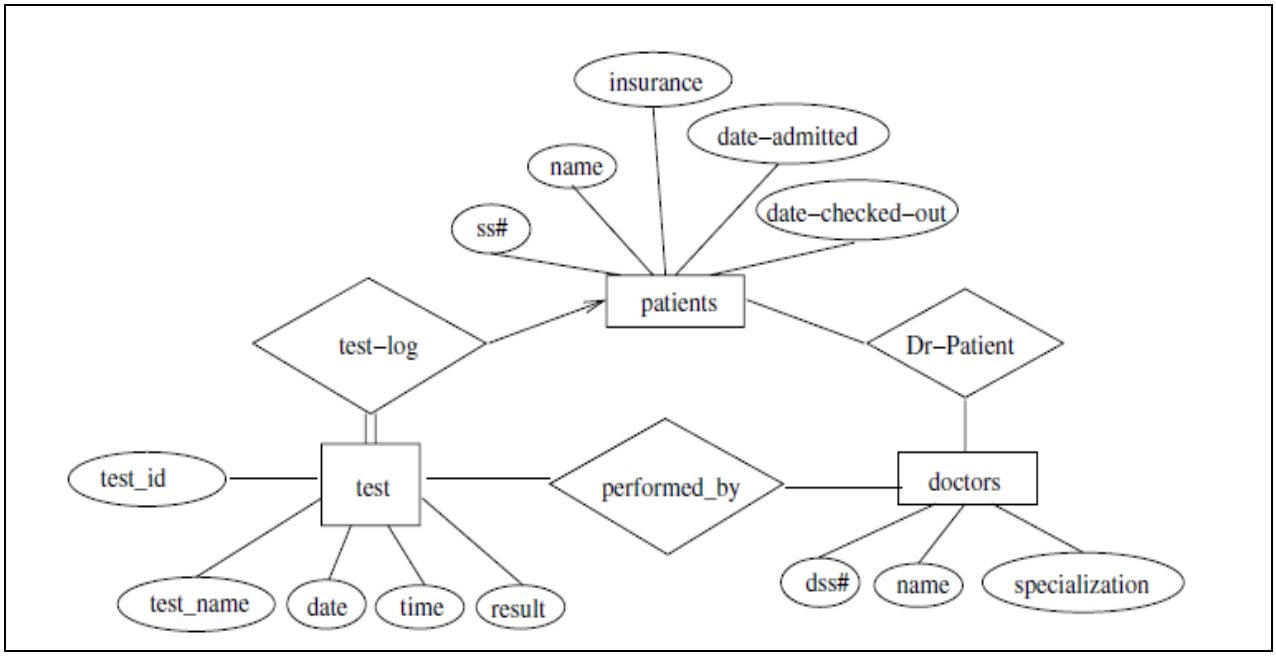
Experiment: 1

Aim: Design of E-R Model.

Procedure: Relate the entities appropriately. Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total /partial). Try to incorporate generalization, aggregation, specialization etc wherever required.

There are some notations which are used to draw E-R MODEL
Entity- Rectangle
Attribute-Oval shape
Relation-Diamond
Strong Entity- Having primary key.
Weak Entity-Not have any primary key.

Example: E-R diagram for a Hospital Management with a set of patients and a set of medical doctors.



Exercises:
1. Draw an E-R diagram for Library Management and Employee Management.
2. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

Experiment: 2

Aim: Study of Relational Database.

Analyze the problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys.

Procedure:

Entity- which is perceived or known or inferred to have its own distinct existence (Living or nonliving)
Attribute- proper ties of entity (column in table)
Tuple- rows in table
Key- set of attributes which can uniquely identify the TUPLE in table.
Candidate Key- Minimal super key
Primary Key- one of the candidate key

Example: Bus Reservation

BUS

BUS_NO	SOURCE	DESTINATION	DAYS_AVAILABLE	BUS_TYPE	VIA	CAPACITY	DEP_TIME	DISTANCE

FARE

BUS_TYPE	DIST_FORM	DIST_TO	ADULT_FARE_PERCENTAGE	CHILD_FARE_PERCENTSGE

RESERVATION

PNR_NO	JOURNEY_DATE	BUS_NO	NO_OF_SEAT	ADDRESS	CONTACT_NO	STATUS	TICKET_NO

CANCELLATION

PNR_NO	JOURNEY_DATE	BUS_NO	NO_OF_SEAT	ADDRESS	CONTACT_NO	STATUS	TICKET_NO

PASSENGER

PSG_NO	NAME	AGE	SEX	ADDRESS	PNR	SEAT_NO	CONTACT_NO	STATUS	TICKET_NO

TICKET

TICKET_NO	BK_DATE	PNR	NO_OF_ADULTS	NO_OF_CHIDREN	TICKET_NO

Exercises:

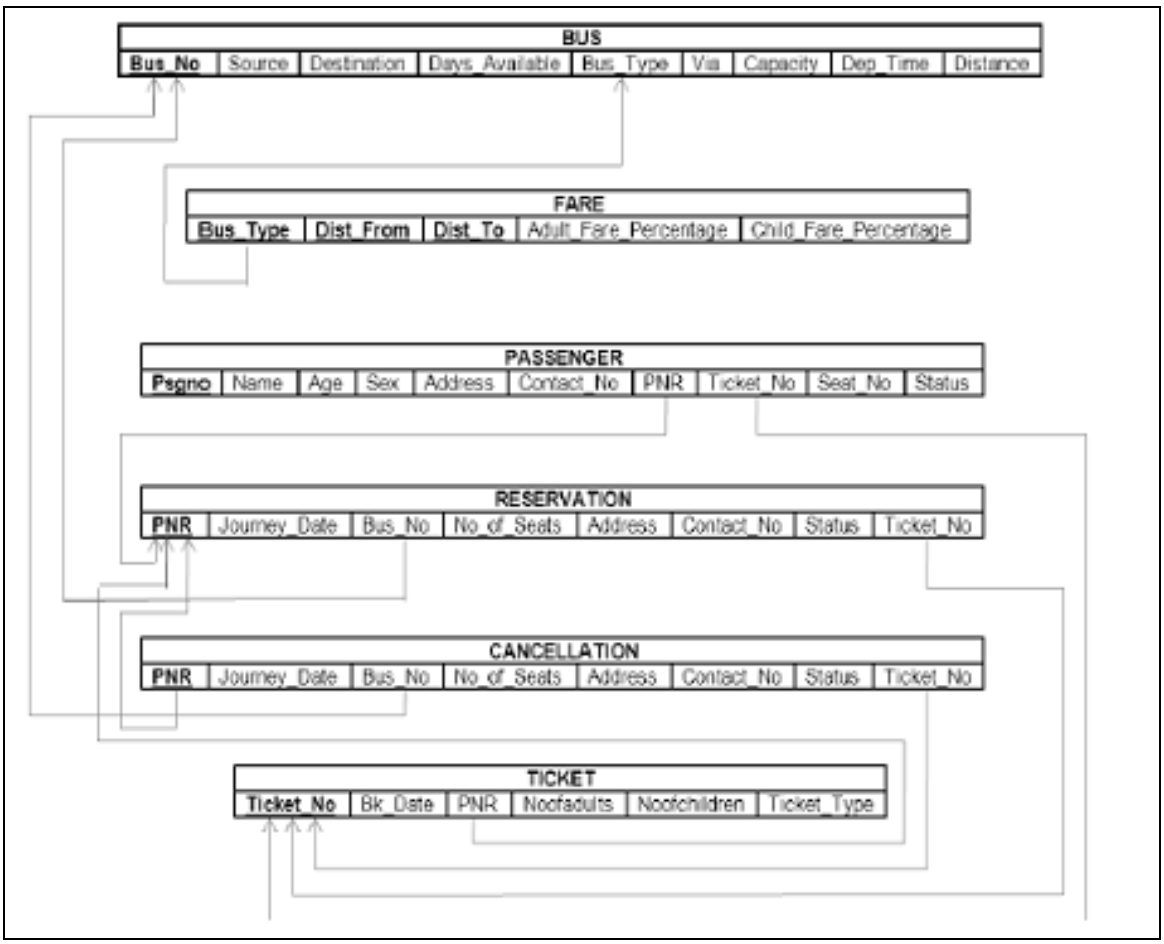
1. Create relational database for Library Management and Employee Management.
2. Create relational database for Hospital Management according to above given E-R model.

Experiment: 3

Aim: Design of Relational Model.

Procedure: Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement. Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

Example: Design Relations between tables of Bus reservation.



Exercises:

1. Draw the relational model for Library management and Employee management.

2. Design a relational database corresponding to the given tables.

person (driver-id, name, address)
car (license, year, model)
accident (report-number, location, date)
owns (driver-id, license)
participated (report-number driver-id, license, damage-amount)
employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)

Experiment: 4

Aim: Write queries of Data Definition Language.

Procedure: DDL is used to perform the operation on schema. By using data definition language we can create table, modify table and delete table. For these we use some query and syntax.

1) To create a table- Create table table name(attribute with data type),
Example:-create table E(eno integer,ename varchar(40),age integer);
2) To alter table- Alter table "table name" Changes;
Example:-a) alter table E
Add column salary integer;
b) alter table E
drop column age;
3) To delete table- Drop table table name;
Example:- drop table E;

Result: **1)** A table is created of name "E" having column eno, ename, age and data type is varchar and variable size max 40.
2) a:-It adds a column in table E" that is salary of integer type
b:-Colume age is delete from table "E"
3) Table "E" is deleted.

Exercises:

Employee Management

1. Write a query to creating Employee_Info table with attribute E_Name, E_Age, E_Sex, E_Address, E_Number, E_Phone, E_Designation, E_Qualification, E_Salary.
2. Write a query to add new attribute (E_section) in Employee_Info table.
3. Write a query to delete attribute (E_Phone) from Employee_Info table.

Library Management

1. Write a query to creating Book_Info table with attribute B_Number, B_Name, B_Cost, B_Publisher, B_Author, B_Noofcopies.
2. Write a query to add new attribute (B_Subject) in Book_Info table.
3. Write a query to delete attribute (B_Author) from Book_Info table.

Experiment: 5

Aim: Write queries of Data Manipulation Language.

Procedure: DML is used to perform the operation on tuple or entry .by using data manipulation language we can insert entry, update entry and delete entry. For these we use some query and syntax.

1) To insert:- Insert into table name alues (--,--,--);
Example:-Insert into table E values (17, kk, 31); or
Insert into table E(eno , age)Values (19,31) ,
2) To update entry:- Update table name Set changes Where condition ;
Example:- update table E Set age =age+1Where eno = 15 ;
3) To delete entry:- Delete from table name Where condition;
Example:- delete From E Where eno = 14;

Result: 1) Row is inserted eno=17, name=kk, age= 31
2) Age is increase by 1 where eno=15
3) All entry of eno= 14 is deleted

Exercises:
Employee Management

1. Write a query to insert information (add row) in Employee_Info table.
2. Write a query to update Employee_Info table so that XYZ employee now lives in ABC.
3. Write a query to delete all information of XYZ employee from Employee_Info table.

Library Management
1. Write a query to insert row in Book_Info table.
2. Write a query to update the value of attribute (B_Subject=DBMS) in Book_Info table.
3. Write a query to delete information of book author from Book_Info table where author=ABC.

Experiment: 6

AIM: Use of Aggregation Function in SQL query.

Procedure: Aggregation function is a command in a database management program that performs an arithmetic operation on the values in a specified column or field in all the records in the database, such as computing their sum or average or counting the number of records that satisfy particular criteria.

There are 5 type of aggregation function.

1) Sum : it calculate sum of particular column
2) Avg : it calculate avg of any column.
3) Count: I t calculate no of tuple in table or no of values in the table.
4) Max: it calculate max value of any column.
5) Min: it calculates min value of any column.

1) SUM: select sum (attribute) from table name;
Example: select sum (salary) from E;
2) AVG: select avg (attribute) from table name
Example: select avg(salary) from E;
3) COUNT: select count (attribute) from table name
Example: select count (*) from E; \\ * use for all entry
4) MAX: select max (attribute) from table name
Example: select max (age) from E;
5) MIN: select min(attribute) from table name
Example: select min (salary) from E;

Result: 1) It will return total salary in table E.
2) It will return average salary in E.
3) It will return total entry in table E.
4) It will return max age of table E.
5) It will return min salary of table E.

Exercises:
Library Management

1. Find the total amount of books from Book_info table.
2. Find the average of book amount from Book_info table.
3. Find the number of book author from book_info table.
4. Find the name of book and author whose cost is minimum.
5. Find the name of book and author whose cost is maximum.

Experiment: 7

Aim: To implement the concept of joins.

Procedure:

Joint Multiple Table (Equi Join): Sometimes we require to treat more than one table as though manipulate data from all the tables as though the tables were not separate object but one single entity. To achieve this we have to join tables .Tables are joined on column that have dame data type and data within tables. The tables that have to be joined are specified in the FROM clause and the joining attributes in the WHERE clause.

Algorithm for JOIN in SQL:

1. Cartesian product of tables (specified in the FROM clause)
2. Selection of rows that match (predicate in the WHERE clause)
3. Project column specified in the SELECT clause.

1. Cartesian Product:
Consider two table student and course
Select B.*,P.* FROM student B, course P;

2. Inner Join:
Cartesian product followed by selection
Select B.*,P.* FROM student B, Course P WHERE B.course # P.course # ;

3. Left Outer Join:
LEFT OUTER JOIN = Cartesian product + selection but include rows from the left table which are unmatched pat nulls in the values of attributes belonging to the second table
Exam: Select B.*,P* FROM student B left join course p ON B.course # P.course #;

4. Right Outer Join:
RIGHT OUTER JOIN = Cartesian product + selection but include rows from right table which are unmatched
Exam: Select B.*,P.* From student B RIGHT JOIN course P B.course# = P course # ;

5. Full Outer Join
Exam: Select B.*,P.* From student B FULL JOIN course P On B.course # = P course # ;

Exercises:
1. Consider the relational database of Bank, where the primary keys are underlined.
employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)
Give an expression in the relational algebra to express each of the following queries:
a. Find the names of all employees who work for First Bank Corporation.
b. Find the names and cities of residence of all employees who work for First Bank Corporation.
c. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.

d. Find the names of all employees in this database who live in the same city as the company for which they work.
e. Find the names of all employees who live in the same city and on the same street as do their managers.
2. Let a and b be relations with the schemas A(name, address, title) and B(name, address, salary), respectively. Show how to express a **natural full outer join** b using the **full outer join** operation with an **on** condition and the **coalesce** operation. Make sure that the result relation does not contain two copies of the attributes name and address, and that the solution is correct even if some tuples in a and b have null values for attributes name or address.

Experiment: 8

Aim: To implement the concept of sub-queries**.**

Procedure: A subquery is a form of an SQL statement that appears inside another SQL statement. It also termed as nested Query. The statement containing a subquery called a parent statement. The rows returned by the subquery are used by the following statement. It can be used by the following commands:
1. To insert records in the target table.
2. To create tables and insert records in this table.
3. To update records in the target table.
4. To create view.
5. To provide values for the condition in the WHERE, HAVING IN, SELECT, UPDATE, and DELETE statements.
Example:- Creating clientmaster table from oldclient_master, table Create table client_master AS SELECT * FROM oldclient_master;

Using the Union, Intersect and Minus Clause:

Union Clause:
The user can put together multiple Queries and combine their output using the union clause. The union clause merges the output of two or more Queries into a single set of rows and column. The final output of union clause will be
Output: = Records only in Query one + records only in Query two + A single set of records with is common in the both Queries.
Syntax: SELECT columnname, columname FROM tablename 1 UNION SELECT columnname, columnname From tablename2;

Intersect Clause: The use can put together multiple Queries and their output using the interest clause. The final output of the interest clause will be:
Output =A single set of records which are common in both Queries
Syntax: SELECT columnname, columnname FROM tablename 1 INTERSECT SELECT columnname, columnname FROM tablename 2;

Minus Clause:- The user can put together multiple Queries and combine their output = records only in Query one
Syntax: SELECT columnname, columnname FROM tablename ;
MINUS SELECT columnname, columnname FROM tablename ;

Exercises:

1. Consider the relational database of Bank (given above). Give an expression in the relational algebra for each request:
   a. Modify the database so that Jones now lives in Newtown.
   b. Give all employees of First Bank Corporation a 10 percent salary raise.
   c. Give all managers in this database a 10 percent salary raise.
   d. Give allmanagers in this database a 10 percent salary raise, unless the salary would be greater than $100,000. In such cases, give only a 3 percent raise.
   e. Delete all tuples in the works relation for employees of Small Bank Corporation.

Experiment: 9

Aim: To implement the concept of indexes and views**.**

Procedure: An index is an ordered list of content of a column or group of columns in a table. An index created on the single column of the table is called simple index. When multiple table columns are included in the index it is called composite index.

Creating an Index for a table:

CREATE INDEX index_name ON tablename(column name);

Composite Index:

CREATE INDEX index_name ON tablename(columnname,columnname);

Creating an UniQuestion Index:

CREATE UNIQUESTION INDEX indexfilename ON tablename(columnname);

Dropping Indexes:

An index can be dropped by using DROP INDEX

DROP INDEX indexfilename;

Views: Logical data is how we want to see the current data in our database. Physical data is how this data is actually placed in our database. Views are masks placed upon tables. This allows the programmer to develop a method via which we can display predetermined data to users according to our desire.

Creation of Views:-

Syntax: CREATE VIEW viewname AS SELECT columnname,columnname FROM tablename WHERE columnname=expression_list;

Renaming the columns of a view:-

Syntax: CREATE VIEW viewname AS SELECT newcolumnname….
FROM tablename WHERE columnname=expression_list;

Selecting a data set from a view-

Syntax: SELECT columnname, columnname FROM viewname WHERE search condition;

Destroying a view-

Syntax: DROP VIEW viewname;

Exercises:

1 Suppose that we have a relation marks(student-id, score) and we wish to assign grades to students based on the score as follows: grade F if score < 40, grade C if 40 ≤ score < 60, grade B if 60 ≤ score < 80, and grade A if 80 ≤ score. Write SQL queries to do the following:

a. Display the grade for each student, based on the marks relation.

b. Find the number of students with each grade.

Experiment: 10

Aim: To implement the concept of grouping of data.

Procedure: Grouping Data from Tables:

There are circumstances where we would like to apply the aggregate function not only to a single set of tuples, but also to a group of sets of tuples, we specify this wish in SQL using the group by clause. The attribute or attributes given in the group by clause are used to form group. Tuples with the same value on all attributes in the group by clause are placed in one group.

SELECT columnname, columnname FROM tablename GROUP BY columnname;

At times it is useful to state a condition that applies to groups rather than to tuples. For example we might be interested in only those branches where the average account balance is more than 1200. This condition does not apply to a single tuple, rather it applies to each group constructed by the GROUP BY clause. To express such Questioners, we use the having clause of SQL. SQL applies predicates in the having may be used.

Syntax:

SELECT columnname, columnname
FROM tablename
GROUP BY columnname;
HAVING searchcondition

Exercises:

1. Consider the relational database of Bank, where the primary keys are underlined.

employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)

Give an expression in the relational algebra to express each of the following queries:

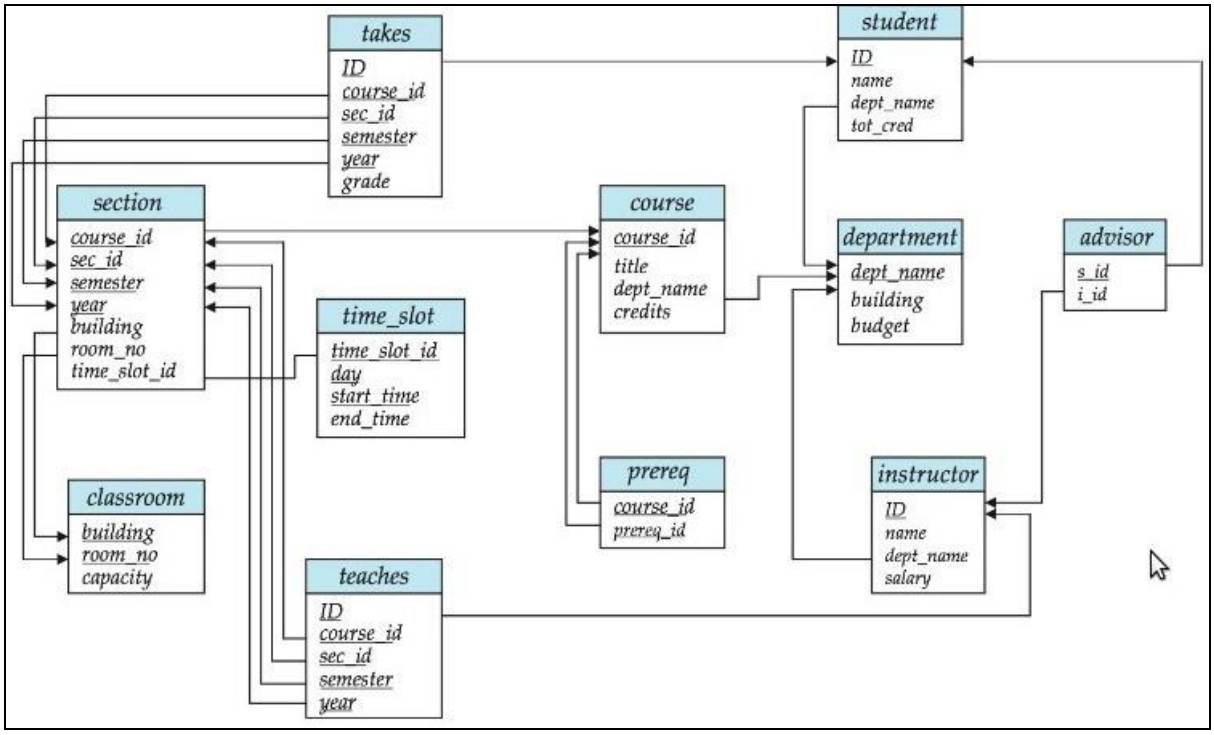
a. Find the names of all employees in this database who do not work for First Bank Corporation.

b. Find the names of all employees who earn more than every employee of Small Bank Corporation.

c. Assume the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.

Practice Questions

Write the following simple SQL Queries on the University Schema:



1. Find the names of all the instructors from Information Technology department
2. Find the course id and titles of all courses taught by an instructor named 'ABC'.
3. Find the ID and name of instructors who have taught a course in the Information Technology department, even if they are themselves not from the Information Technology department.
4. Find the courses which are offered in both 'Fall' and 'Spring' semester.
5. Find names of instructors who have taught at least one course in Spring 2009

Aggregates and grouping, and ordering

1. Find the number of instructors who have never taught any course. If the result of your query is 0, add appropriate data (and include corresponding insert statements) to ensure the result is not 0.
2. Find the total capacity of each of the buildings in the university.
3. Find the maximum number of teachers for any single course section.
4. Find all departments that have the minimum number of instructors, using a subquery; order the result by department name in descending order.
5. For each student, compute the total credits they have successfully completed, i.e. total credits of courses they have taken, for which they have a non-null grade other than 'F'. Do NOT use the total credit attribute of student.
6. Find the number of students who have been taught (at any time) by an instructor named 'ABC'. Make sure you count a student only once even if the student has taken more than one course from ABC.
7. Find the name of all instructors who get the highest salary in their department.
8. Find all students who have taken all courses taken by instructor 'ABC'.
9. Find the total money spent by each department for salaries of instructors of that department.
(a) First write the query ignoring the case of departments with no instructor.
(b) Modify your query to ensure departments with no instructors are output with a value of 0.
10. List out the names of all Students whose advisor teaches maximum no of subjects.

Viva Questions

1. What is database?
2. What is DBMS?
3. Disadvantage in File Processing System?
4. Define the "integrity rules"
5. What is extension and intension?
6. What is a view? How it is related to data independence?
7. What is an Entity type?
8. Describe Weak Entity set?
9. What is an attribute?
10. What is a Relation Schema and a Relation?
11. What is degree of a Relation?
12. Define Relationship set?
13. What is DML (Data Manipulation Language)?
14. What is VDL (View Definition Language)?
15. What is DML Compiler?
16. What is DDL Interpreter?
17. Define normalization?
18. What is Functional Dependency?
19. How to find out the database name from SQL*PLUS command prompt?
20. What is the difference between SQL and SQL Server?
21. What is difference between Co-related sub query and nested sub query?
22. What is difference between Oracle and MS Access?
23. What are disadvantages in Oracle and MS Access?
24. What are features & advantages in Oracle and MS Access?
25. What is difference between DBMS and RDBMS?
26. What are the advantages and disadvantages of primary key and foreign key in SQL?
27. Which date function is used to find the difference between two dates?

References:

1. Elmasri, Navathe, “Fundamentals Of Database Systems”, Addision Wesley.
2. Korth, Silbertz, Sudarshan, “Database Concepts”, McGraw Hill.
3. Date C J, “An Introduction to Database System”, Addision Wesley.
4. Leon & Leon, Fundamental of Data Base Management System, TMH.
5. Oracle 9i Database Administration fundamental – I, volume 1, Oracle Press.