****

**UNIVERSITY COLLEGE OF ENGINEERING**

**(BIT CAMPUS), TIRUCHIRAPPALLI-620 024**

**First Internal Assessment Test**

**Department of CSE / IT**

**Subject Code / Name:** CS8492 / Database Management Systems

**Date Duration :** 01.02.2019 & 1.30Hrs **Marks: 50**

**Degree/Branch:** B.TECH/IT Sec’B’ **Year / Sem:** II / IV

**ANSWER KEY**

**PART-A**

**Answer *All* Questions 7 x 2 = 14**

1. **Who is a DBA? What are the responsibilities of a DBA?**

A database administrator (short form DBA) is a person responsible for the design, implementation, maintenance and repair of an organization's database. They are also known by the titles Database Coordinator or Database Programmer, and is closely related to the Database Analyst, Database Modeller, Programmer Analyst, and Systems Manager.The role includes the development and design of database strategies, monitoring and improving database performance and capacity, and planning for future expansion requirements. They may also plan, co-ordinate and implement security measures to safeguard the database.

1. **What is a data model? List the types of data model used.**

A database model is the theoretical foundation of a database and fundamentally determines in which manner data can be stored, organized, and manipulated in a database system. It thereby defines the infrastructure offered by a particular database system. The most popular example of a database model is the relational model.

Types of data model used

 Hierarchical model

 Network model

 Relational model

 Entity-relationship

 Object-relational model

 Object model

1. **What are the disadvantages of file processing system?**

The disadvantages of file processing systems are

a) Data redundancy and inconsistency

b) Difficulty in accessing data

c) Data isolation

d) Integrity problems

e) Atomicity problems

f) Concurrent access anomalies

1. **Define instance and schema.**

**Instance**: Collection of data stored in the data base at a particular moment is called an Instance of the database.

**Schema**: The overall design of the data base is called the data base schema.

1. **What is relationship? Give examples.**

A relationship is an association among several entities.

**Example**: A depositor relationship associates a customer with each account that he/she has.

1. **Define Super Key, Candidate Key and Primary Key.**

Definition of Candidate Key in DBMS: A super key with no redundant attribute is known as candidate key. Candidate keys are selected from the set of super keys, the only thing we take care while selecting candidate key is that the candidate key should not have any redundant attributes.

1. **Define weak and strong entity sets?**

**Weak entity** set: entity set that do not have key attribute of their own are called weak entity sets.

**Strong entity set**: Entity set that has a primary key is termed a strong entity set.

1. **Explain the two types of participation constraint.**

**Total:** The participation of an entity set E in a relationship set R is said to be total if every entity in E participates in at least one relationship in R.

**Partial:** if only some entities in E participate in relationships in R, the participation of entity set E in relationship R is said to be partial**.**

1. **What is meant by functional dependencies?**

Consider a relation schema R and a C R and ß C R. The functional dependency a ß holds on relational schema R if in any legal relation r(R), for all pairs of tuples t1 and t2 in r such that t1 [a] =t1 [a], and also t1 [ß] =t2 [ß].

1. **What are all the characteristics of properly normalized database?**

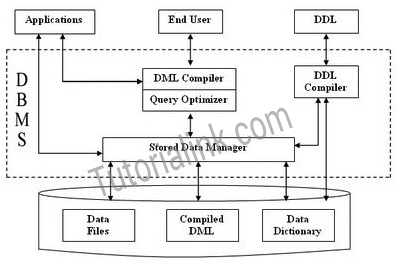
It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

* Minimizing redundancy
* Minimizing insertion, deletion and updating anomalies

**PART-B**

**Answer any three Questions 3 x 10 = 30**

1. **Explain briefly about all the components in database system architecture (Structures) with a neat diagram.**



Components of a DBMS

* DDL Compiler: Data Description Language compiler processes schema definitions specified in the DDL. ...
* DML Compiler and Query optimizer: ...
* Data Manager: ...
* Data Dictionary: ...
* Data Files: ...
* Compiled DML: ...
* End Users:

1. **Write short notes on**
2. **Embedded SQL**

Embedded SQL is a method of combining the computing power of a programming language and the database manipulation capabilities of SQL. Embedded SQL statements are SQL statements written inline with the program source code , of the host language. The embedded SQL statements are parsed by an embedded SQL preprocessor and replaced by host-language calls to a code library. The output from the preprocessor is then compiled by the host compiler. This allows programmers to embed SQL statements in programs written in any number of languages such as C/C++, COBOL and Fortran. This differs from SQL-derived programming languages that don't go through discrete preprocessors, such as PL/SQL and T-SQL.

The SQL standards committee defined the embedded SQL standard in two steps: a formalism called Module Language was defined, then the embedded SQL standard was derived from Module Language.[1] The SQL standard defines embedding of SQL as embedded SQL and the language in which SQL queries are embedded is referred to as the host language. A popular host language is C. Host language C and embedded SQL, for example, is called Pro\*C in Oracle and Sybase database management systems and ECPG in the PostgreSQL database management system. SQL may also be embedded in languages like PHP etc.

1. **Dynamic SQL**

Dynamic SQL is SQL statements that are constructed at runtime; for example, the application may allow users to enter their own queries. Dynamic SQL is a programming technique that enables you to build SQL statements dynamically at runtime.

1. **What is Normalization? Explain in details about 1NF, 2NF, 3NF and BCNF with suitable examples.**

Database Normalization is a technique of organizing the data in the database. Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anamolies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables.

* Normalization is used for mainly two purposes,
* Eliminating reduntant(useless) data.

**Normalization Rule**

Normalization rules are divided into the following normal forms:

* First Normal Form
* Second Normal Form
* Third Normal Form
* BCNF
* Fourth Normal Form

**First Normal Form (1NF)**

For a table to be in the First Normal Form, it should follow the following 4 rules:

It should only have single(atomic) valued attributes/columns.

Values stored in a column should be of the same domain

All the columns in a table should have unique names.

And the order in which data is stored, does not matter.

In the next tutorial, we will discuss about the First Normal Form in details.

**Second Normal Form (2NF)**

For a table to be in the Second Normal Form,

It should be in the First Normal form.

And, it should not have Partial Dependency.

To understand what is Partial Dependency and how to normalize a table to 2nd normal for, jump to the Second Normal Form tutorial.

**Third Normal Form (3NF)**

A table is said to be in the Third Normal Form when,

It is in the Second Normal form.

And, it doesn't have Transitive Dependency.

Here is the Third Normal Form tutorial. But we suggest you to first study about the second normal form and then head over to the third normal form.

**Boyce and Codd Normal Form (BCNF)**

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

R must be in 3rd Normal Form

and, for each functional dependency ( X → Y ), X should be a super Key.

1. **For the following relation schema: (Write Any five queries)**

**employee(employee-name, street, city)**

**works(employee-name, company-name, salary)**

**company(company-name, city)**

**manages(employee-name, manager-name)**

Give an expression in SQL for each of the following queries:

1. Find the names, street address, and cities of residence for all employees who work for 'First Bank Corporation' and earn more than $10,000.

select employee.employee-name, employee.street, employee.city from employee, works where employee.employee-name=works.employee-name and company-name = 'First Bank Corporation' and salary > 10000)

1. b) Find the names of all employees in the database who live in the same cities as the companies for which they work.

select e.employee-name from employee e, works w, company c where e.employee-name = w.employee-name and e.city = c.city and w.company-name = c.company-name

1. Find the names of all employees in the database who live in the same cities and on the same streets as do their managers.

select p.employee-name from employee p, employee r, manages m where p.employee-name = m.employee-name and m.manager-name = r.employee-name and p.street = r.street and p.city = r.city

1. Find the names of all employees in the database who do not work for 'First Bank Corporation'. Assume that all people work for exactly one company.

select employee-name from works where company-name <> 'First Bank Corporation'

1. Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'. Assume that all people work for at most one company.

select employee-name from works where salary > all (select salary from works where company-name = 'Small Bank Corporation')

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

select s.company-name from company s where not exists  
((select city from company where company-name = 'Small Bank Corporation') except (select city from company t where s.company-name = t.company-name))

1. Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.

select employee-name from works t where salary >(select avg(salary) from works s where t.company-name = s.company-name)

1. Find the name of the company that has the smallest payroll.

select company-name from works group by company-name having sum(salary) <= all (select sum(salary)  
from works group by company-name)