Q.2 a. Explain the different roles of database administrators.

Answer:

Roles of database administrators:

1. Planning for the database's future storage requirements
2. Defining database availability and fault management architecture
3. Defining and creating environments for development and new release installation
4. Creating physical database storage structures after developers have designed an application
5. Constructing the database
6. Determining and setting the size and physical locations of data files
7. Evaluating new hardware and software purchase
8. Researching, testing, and recommending tools for Oracle development, modeling, database administration, and backup and recovery implementation, as well as planning for the future
9. Providing database design and implementation
10. Understanding and employing the optimal flexible architecture to ease administration, allow flexibility in managing I/O, and to increase the capability to scale the system

b. How Specialization is differing from generalization? Explain with the suitable example.

Answer: Refer Text Book 4.2.1 & 4.2.2 page No. 106-109

c. Define the following terms associated with ER Model:
(i) Simple and Composite Attribute
(ii) Single Valued & Multi valued attribute
(iii) Primary Key and Foreign Key

Answer:

(i) Simple and Composite Attribute

SIMPLE attributes are attributes that are drawn from the atomic value domains E.g. Name = {John} ; Age = {23}
COMPOSITE attributes: Attributes that consist of a hierarchy of attributes E.g. Address may consists of “Number”, “Street” and “Suburb” _ Address = {59 +‘Meek Street’ + ‘Kingsford’}

(ii) Single Valued & Multi valued attribute
SINGLE VALUED attributes: Attributes that have only one value for each entity E.g. Name, Age for EMPLOYEE
MULTIVALUED attributes: Attributes that have a set of values for each entity E.g. Degrees of a person: ‘ BSc’ , ‘MIT’, ‘PhD’

(iii) Primary Key and Foreign Key

Primary Key: The primary key is an attribute or a set of attributes that uniquely identify a specific instance of an entity. Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.

To qualify as a primary key for an entity, an attribute must have the following properties:

* It must have a non-null value for each instance of the entity.
* The value must be unique for each instance of an entity
* The values must not change or become null during the life of each entity instance.

Foreign Key: The primary key of one files or table which is implanted in another file or table to implement the relationships between them. Foreign keys are used to implement some types of relationships. Foreign keys do not exist in information models..

Q.3 a. What do you mean by Referential Integrity? Define it in the terms of:
(i) Referential Integrity in the ER Model
(ii) Referential Integrity in SQL

Answer:

(i) **Referential Integrity in the ER Model**
Page No. 157-158 section 5.2.4 of Text Book

(ii) **Referential Integrity in SQL –** In referential integrity, it is ensured that a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation.

In SQL, entity integrity and referential integrity rules are implemented as constraints on the relation called as primary key constraint and reference key constraint respectively. These constraints can be specified with relation at the time of creation of the relations or after the creation of the relations by altering the definition of the relations. For example:

```
CREATE TABLE DEPT
(DEPTNO NUMBER PRIMARY KEY,
DNAME VARCHAR2(15));
CREATE TABLE EMP
(EMPNO NUMBER PRIMARY KEY,
ENAME VARCHAR2(15),
JOB VARCHAR2(10),
DEPTNO NUMBER REFERENCES DEPT(DEPTNO));
```

b. What is the difference between Natural Join and Outer Join? Explain with the help of Relational Algebra.

Answer:

Natural join is a binary operator that is written as (*R* * *S*) where *R* and *S* are relations. The result of the natural join is the set of all combinations of tuples in *R*

and *S* that are equal on their common attribute names. In this only one column out of columns having same name attributes is retained.

An **Outer join** contains those tuples and additionally some tuples formed by extending an unmatched tuple in one of the operands by "fill" values for each of the attributes of the other operand.

Relational Algebra: Refer Page No. 191 of Text Book

Q.4 a. A Database Schema is defined as:

Engineer (EngineerID, eName, DOB, Designation, Income, Dept_ID)
PROJECT (Project_No., ProjectName, Budget, Dept_ID)
DEPARTMENT (Dept_ID, DeptName, MGR EngineerID)
WorksOn (EngineerID, Project_No., Duty, Hours)

Based on the above, answer the following Questions:

(i) Write an SQL query that returns the Engineers (IDs and Name only) who have a title of 'SWD' or 'SWT' and earn more than Rupees Sixty Five Thousands.

(ii) Write an SQL query that returns the Engineers (name only) in department 'SWD 2 ' ordered by decreasing income.

(iii) Write an SQL query that returns the Engineer Name, Department Name, and Engineer Designation.

(iv)Write an SQL query that returns the Engineer IDs and Incomes of all Engineers in the 'SW Testing' department ordered by descending income.

(v)Write an SQL query that returns the Engineer name, Project name, Engineer Designation and Hours for all Works On records.

Answer:

(i)SELECT Engineer_ID, eName
FROM Engineer
WHERE (title = 'SWD' OR title = 'SWT') AND income > 65000

(ii) SELECT eName
FROM Engineer
WHERE Dept_ID = 'SWD 2 '
ORDER BY income DESC

(iii) SELECT eName, DeptName, Designation
FROM Engineer, DEPARTMENT
WHERE Engineer. Dept_ID = DEPARTMENT. Dept_ID

(iv) SELECT Engineer_ID, income
FROM Engineer, DEPARTMENT
WHERE DeptName = 'SW Testing' and Engineer.Dept_ID =
DEPARTMENT.Dept_ID
ORDER BY income DESC

(v) SELECT eName, ProjectName , Designation, Hours
FROM Engineer, PROJECT, WorksOn
WHERE Engineer.EngineerID = WorksOn. EngineerID and PROJECT.
Project_No. = WorksOn. Project_No.

b. List out the main approaches to database programming. What are the advantages and disadvantages of each approach?

Answer: Page No. 303 Section 9.1.1 of Text Book

Q.5 a. Define the concept of Functional Dependencies. List out the main characteristics of functional dependencies that are used when normalizing a relation.

Answer:

Functional dependency describes the relationship between attributes in a relation. For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.

Functional dependency is a property of the meaning or semantics of the attributes in a relation. The semantics indicate how the attributes relate to one another and specify the functional dependencies between attributes. When a functional dependency is present, the dependency is specified as a constraint between the attributes

The main characteristics of functional dependencies that we use in normalization

- have a one-to-one relationship between attribute(s) on the left and right-hand side of a dependency;
- hold for all time;
- are nontrivial.

b. Explain in detail the 1NF (first normal form), 2NF (second normal form) and 3NF (third normal form). Also give suitable examples for explanation.

Answer:

1. First normal form (1NF) is now considered to be part of the formal definition of a relation in the basic (flat) relational model; it was defined to disallow multivalued attributes, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic (simple, indivisible) values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. Hence, 1NF disallows having a set of values, a tuple of values, or a combination of both as an attribute value for a single tuple. In other words, 1NF disallows "relations within relations" or "relations as attributes of tuples." The only attribute values permitted by 1NF are single atomic (or indivisible) values.

Consider the DEPARTMENT relation schema shown in Figure1, whose primary key is DNUMBER, and suppose that we extend it by including the DLOCATIONS attribute as shown in Figure 2(a). We assume that each department can have a number of locations. The DEPARTMENT schema and an example extension are shown in Figure 2. As we can see, this is not in 1NF because DLOCATIONS is not an atomic attribute, as illustrated by the first tuple in Figure 2(b). There are two ways we can look at the DLOCATIONS attribute:

• The domain of DLOCATIONS contains atomic values, but some tuples can have a set of these values. In this case, DLOCATIONS is not functionally dependent on DNUMBER.

2. Second normal form (2NF) is based on the concept of full functional dependency. A functional dependency Xâ Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more; that is, for any attribute A X, (X - {A}) does not functionally determine Y. A functional dependency X â Y is a partial dependency if some attribute A X can be removed from X and the dependency still holds; that is, for some A X, (X - {A}) â Y. In Figure, {SSN, PNUMBER}âHOURS is a full dependency (neither SSNâHOURS nor PNUMBERâ HOURS holds). However, the dependency {SSN, PNUMBER} âENAME is partial because SSNâENAME holds. The test for 2NF involves testing for functional dependencies whose left-hand side attributes are part of the primary key. If the primary key contains a single attribute, the test need not be applied at all. A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

3. Third Normal Form: Third normal form (3NF) is based on the concept of transitive dependency. A functional dependency Xâ Y in a relation schema R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key of R, and both Xâ Z and ZâY hold. According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

Figure 1 Simplified version of the COMPANY relational database schema.

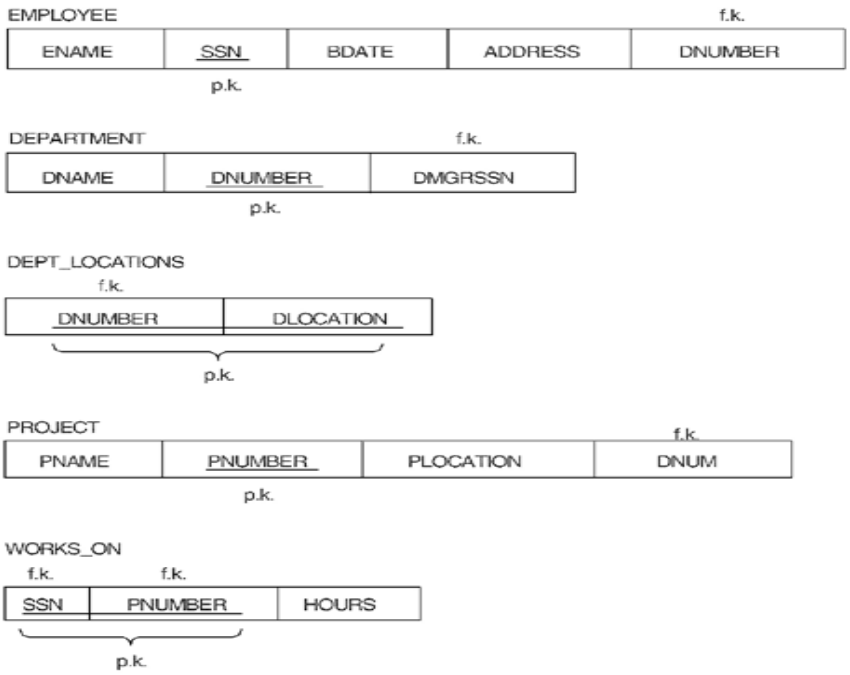
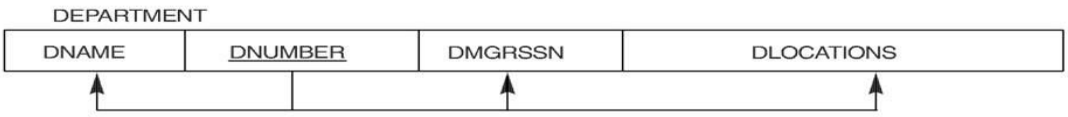


Figure Normalization into 1NF. (a) Relation schema that is not in 1NF. (b) Example relation instance. (c) 1NF relation with redundancy.

(a)



(b) DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c) DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

c. What do you understand by multi-valued dependency? How this concept relates to 4NF?

Answer:

Multivalued dependencies are a consequence of first normal form (1NF), which disallowed an attribute in a tuple to have a set of values. If we have two or more multivalued independent attributes in the same relation schema, we get into a problem of having to repeat every value of one of the attributes with every value of the other attribute to keep the relation state consistent and to maintain the independence among the attributes involved. This constraint is specified by a multivalued dependency. For example, consider the relation EMP. A tuple in EMP relation represents the fact that an employee whose name is ENAME works on the project whose name is PNAME and has a dependent whose name is DNAME. An employee may work on several projects and may have several dependents, and the employee's projects and dependents are independent of one another. To keep the relation state consistent, we must have a separate tuple to represent every combination of an employee's dependent and an employee's project. This constraint is specified as a multivalued dependency on the EMP relation. Informally, whenever two *independent* 1:N relationships *A:B* and *A:C* are mixed in the same relation, an MVD may arise.

Multi-valued Dependency (MVD) Represents a dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. However, the set of values for B and C are independent of each other.

Fourth Normal Form is violated when a relation has undesirable multivalued dependencies, and hence can be used to identify and decompose such relations. A relation schema *R* is in 4NF with respect to a set of dependencies *F* (that includes functional dependencies and multivalued dependencies) if, for every *nontrivial* multivalued dependency *X Y* in, *X* is a superkey for R. The EMP relation is not in 4NF because in the nontrivial MVDs ENAME PNAME and ENAME DNAME, ENAME is not a superkey of EMP. We decompose EMP into EMP_PROJECTS and EMP_DEPENDENTS, .Both EMP_PROJECTS and EMP_ DEPENDENTS are in 4NF, because the MVDs ENAME PNAME in EMP_PROJECTS and ENAME DNAME in EMP_DEPENDENTS are trivial MVDs. No other nontrivial MVDs hold in either EMP_PROJECTS or EMP_DEPENDENTS. No FDs hold in these relation schemas either.

Fourth Normal Form (4NF) A relation that is in Boyce-Codd Normal Form and contains no nontrivial multi-valued dependencies.

Fourth normal form (4NF) is a stronger normal form than BCNF as it prevents relations from containing nontrivial MVDs, and hence data redundancy. The normalization of BCNF relations to 4NF involves the removal of the MVD from the relation by placing the attribute(s) in a new relation along with a copy of the determinant(s).

Q.6 a. Briefly describe the following:
(i) Distribution Transparency
(ii) Fragmentation Transparency
(iii) Replication Transparency

Answer:

(i) **Page No. 879 Section 25.1.2 to Text Book**

(ii) **Page No. 879 Section 25.1.2 to Text Book**

(iii) **Page No. 879 Section 25.1.2 to Text Book**

b. What do you mean by data replication and allocation in DDBMS? Explain by appropriate example.

Answer: Page No. 885-888 Section 25.2.2 to Text Book

c. List out the problems occurred in the distributed DBMS for concurrency control and recovery purposes.

Answer: Page No. 897-898 Section 25.5 of Text Book

Q.7 a. What are the techniques for concurrency control? Explain in brief two phase locking technique.

Answer: Page No. 644-654 Section 18.1 to Text Book

b. Explain the following terms with significant examples:
(i) A read only transaction
(ii) An aborted transaction

Answer:

(i) Page No. 613-14 Section 17.1.2 to Text Book
(ii) Page No. 619 Section 17.2.1 to Text Book

c. What do you understand by the terms deadlock and starvation in database transaction? Explain the different approaches to dealing these problems.

Answer: Page No. 651-654 Section 18.1.3 to Text Book

Q.8 a. Explain the UNDO/REDO and UNDO/NO-REDO algorithms for recovery with immediate update. Develop an outline or procedure for an UNDO/REDO recovery algorithm Based on Immediate update for a Multiuser environment.

Answer: Page No. 682-684 Section 19.3, 19.3.2 to Text Book

b. Describe the write-ahead logging protocol. Also mention the actions taken by the recovery manager during check-pointing.

Answer: Page No. 675-676 Section 19.1.3 and 19.1.4 to Text Book

Q.9 a. Distinguish between discretionary access control (DAC) and mandatory access Control (MAC).

Answer: Refer Text Book section 23.3.1 page No. 809-810

b. List out the types of privileges at the account level and those at the relation level.

Answer: Refer Text Book section 23.2.1 page No. 802-803

c. What is the goal of encryption? Give an example of an encryption algorithm and explain how it works.

Answer: Refer Text Book section 23.6 page No. 816-818

TEXT BOOK

I. Fundamentals of Database Systems, Ramez Elmasri, Shamkant B. Navathe, 5th Edition, Pearson Education, 2008