DATABASE MANAGEMENT SYSTEM

For
COMPUTER SCIENCE

DATABASE MANAGEMENT SYSTEM

SYLLABUS

ER model. Relational model: relational algebra, tuple calculus, SQL. Integrity constraints, normal forms. File organization, indexing (e.g., B and B+ trees). Transactions and concurrency control.

ANALYSIS OF GATE PAPERS

Exam Year	1 Mark Ques.	2 Mark Ques.	Total
2003	2	3	8
2004	2	5	12
2005	3	4	11
2006	1	4	9
2007	-	6	12
2008	1	5	11
2009	-	5	10
2010	2	2	6
2011	-	2	4
2012	2	5	12
2013	-	3	6
2014 Set-1	2	3	8
2014 Set-2	2	3	8
2014 Set-3	2	3	8
2015 Set-1	2	2	6
2015 Set-2	2	2	6
2015 Set-3	2	2	6
2016 Set-1	4	1	6
2016 Set-2	2	2	6
2017 Set-1	2	3	8
2017 Set-2	2	4	10
2018	3	2	7

CONTENTS

1 THE ENTITY – RELATIONSHIP MODEL

1.1 INTRODUCTION

An entity is any object, place or activity about which an enterprise keeps data. It is an object which can have instances or occurrences. An entity type is a set of objects which share common properties.
The Database Design consists of three components: Conceptual Design on the basis of user requirements, Data Modeling (Entity – Relationship Diagrams and Normalization), and Physical Design, and Implementation. The major step in conceptual design is to identify entities and relationships, which reflect the data in a natural way. The aim of this step is to specify the conceptual structure of the data. This is known as data modeling. The Entity – Relationship (E – R) model is used as an information model to develop conceptual structure.

1.2 THE E – R MODEL

The E – R data model considers the real world consisting of a set of basic objects and relationships among these objects. A number of attributes are associated with an entity and the attributes describing it. The set of all entities or relationships of the same type is called the entity set or relationship set.

1.3 DATABASE ADMINISTRATOR (DBA)

All controlling of a database system is done by database administrator.

1.3.1 FUNCTIONS OF DBA

1) Decides the storage structure and access strategy.
2) Creation of data dictionary for statistical analysis.
3) Responding to changes in requirements.
4) Performance monitoring.
5) Strategy design for backup and recovery.
6) Authorization checks and validation procedures.
7) Decides the information content.

1.4 THE ENTITY RELATIONSHIP DIAGRAM

1.4.1 RELATIONSHIPS AND RELATIONSHIP SETS

A relationship expresses an association between entities. A relationship set is a set of relationships of the same type. A relationship may also have descriptive attributes. For example, data (last data of account access) could be an attribute of the relationship set.

1.4.1.1 MAPPING CARDINALITIES

It indicates the number of entities with which another entity can be associated via a relationship. The degree of relationship is called cardinality.

a) One-to-one: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.

b) One-to-many: An entity in A is associated with any number in B. An entity in B is associated with at least one entity in A.

c) Many-to-one (N:1): An entity in A is associated with at most one entity in B. An entity in B is associated with any number in A.

1.4.2 KEYS

Differences between entities must be expressed in terms of attributes known as keys. These facilitate us to uniquely identify each entity in a set.

1.4.3 SUPER KEY

It is a set of one or more attributes which put together enable us to identify uniquely an entity in the entity set.

1.4.4 CANDIDATE KEY

A super key may contain extraneous attributes, and we are often interested in the smallest super key. A super key for which no subset is a super key is called a candidate key.

1.4.5 PRIMARY KEY

It is a candidate key (there may be more than one) chosen by the database designer to identify entities in an entity set. The idea of strong and weak entity sets is related to the existence dependencies such as the member of a strong entity set is a dominant entity, and the member of a weak entity set is a subordinate entity. A weak entity set does not have a primary key, but we need a means of distinguishing among the entities.

1.4.6 GENERALIZATION

Generalization hides differences and emphasizes similarities. Distinction is made through attribute inheritance.
Attributes of higher – level entity are inherited by lower - level entities.

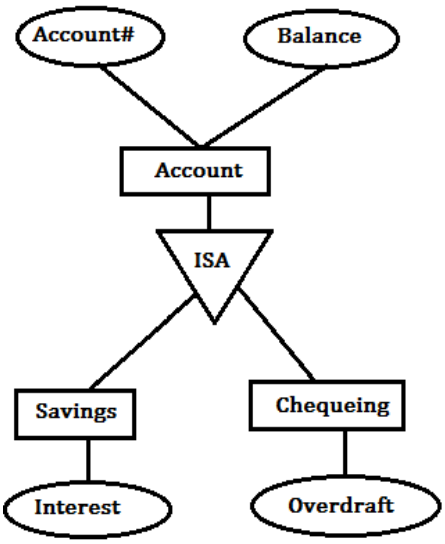


Figure : Generalization

1.4.7 AGGREGATION

The E – R model cannot express relationships among relationships. When would we need such a thing?
Consider a database with information about employees who work on a particular project and using a number of machines for doing that work.
We get the E-R diagram shown in Figure.

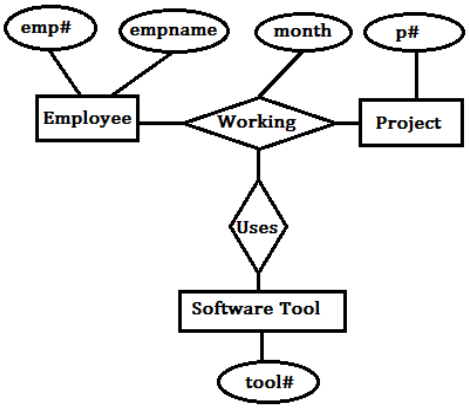


Figure : E - R diagram with redundant relationships

Relationship sets Working and Uses could be combined into a single set. However, they shouldn't be, as this would obscure the logical structure of this scheme. The solution is to use aggregation. It is an abstraction through which relationships are treated as higher – level entities. For our example, we can treat the relationship set Working and the entity sets Employee and Project as a higher –level entity set called Working. Following figure shows the E-R diagram with aggregation.

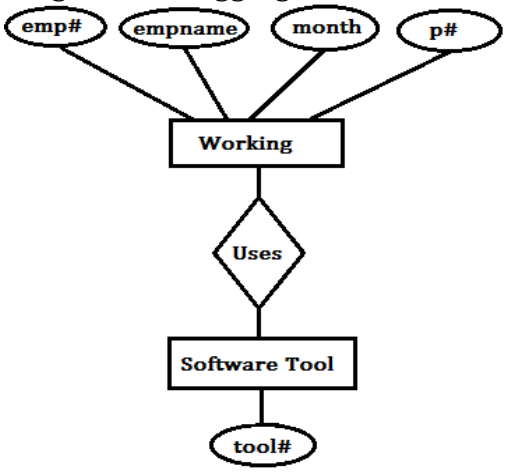


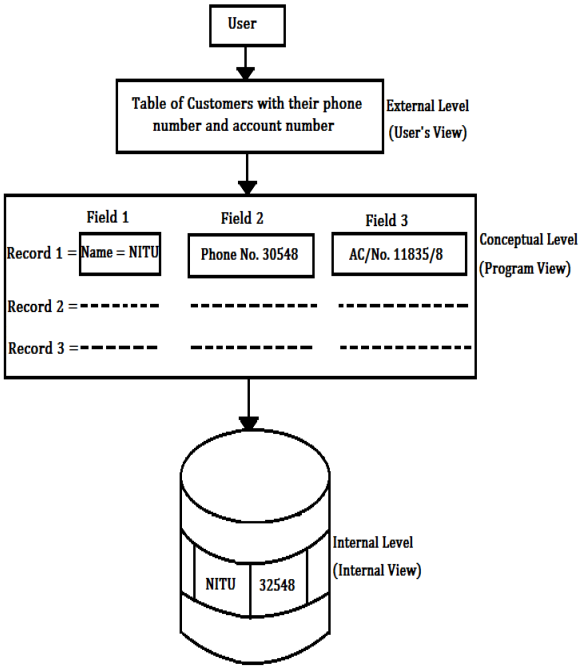
Figure : E - R diagram with aggregation

Transforming an E-R diagram with aggregation into tabular form is easy. We can create a table for each entity and relationship set as before. The table for relationship set Uses contains a column for each attribute in the primary key of Software Tool and Working.

2 RELATIONAL MODEL

2.1 LEVELS OF A DATABASE SYSTEM

The relational model represents the database as a collection of relations. Informally, each relation resembles at stable of values or, to some extent, a "flat" file of records



When a relation is thought of as a table of values, each row in the table represents a collection of related data values. In the relational model, each row in the table represents a fact that typically corresponds to real- world entity or relationship. The table name and column names are used to help Number, Class, Major –specify how to interpret the data values in each row, based on the column each value is in. All values in a column are of the same data type. In the formal relational mode terminology, a row is called a tuple, a column header is called an attribute, and the table is called a relation. The data type describing the types of values that can appear in each column. We now define these terms-domain, tuple, attribute, and relation- more precisely.

2.1.1 DOMAINS, ATTRIBUTES, TUPLE & RELATIONS

A domain D is a set of atomic values. By automatic we mean that each value in the domain is indivisible as far as the relational model is concerned. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a for the domain, to help in interpreting its values.

Some examples of domains

- **USA – phone – numbers:** The set of 10-digit phone numbers valid in the United States.
- **Social –security – numbers:** The set of valid 9 – digit social security numbers.
- **Employee – ages:** Possible ages of employees of a company; each must be a value between 15 and 80 years old.
- **Academic – department – names:** The set of academic department names, such as computer Science, Economics, and Physics, in a university.

Academic –department – codes: The set of academic department codes, such as CS, ECON and PHYS, in a university.

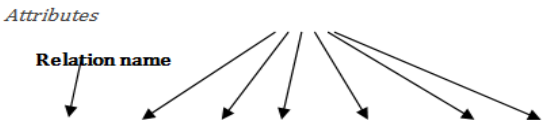
The preceding is called logical definition of domains. A data type or format is also specified for each domain.

e. g. the data type for the domain USA-phone – numbers can be declared as a character string of the form (ddd)ddd – dddd, where each d is a numeric (decimal) digit and the first three digits form a valid telephone area code. The data type for Employee – ages is an integer number between 15 and 80. For Academic – department – names, the data type is the set of all character strings that represents valid department names. A domain is thus given a name, data type, and format. Additional information for interpreting the

values of a domain can also be given; for example, a numeric domain such as Person-weights should have the units of measurement – pounds or kilograms. A relation scheme R, denoted by R(A_1, A_2,…….A_n), is made of a relation name R and a list of attributes A_1, A_2,……..A_n. Each attribute A_1 is the name of a role played by some domain D in the relation schema R. D is called the **domain** of A_i and is denoted by dom(A_i)

A relation schema is used to describe a relation; R is a called the name of this relation. The degree of a relation is the number of attributes n of its relation schema.

An example of a relation schema for a relation of degree 7, which describes university students, is given below STUDENT (Name, SSN, Homophone, Address, Office Phone, Age, GPA)



Student	Name	SSN	Home Phone	Address	Office Phone	GPA
Tuples	Benjamin Bayer	205-61-2435	373-1616	2918 Blue Bonnet lane	null	3.21
	Katherine Ashley	381-62-1245	375-4409	125 Kirby Road	null	2.89
	Dick Davidson	422-22-2320	Null	3452 Elgin Road	748-1253	3.53
	Charles cooper	489-22-1100	376-9821	265 Lark lane	749-6492	3.93
	Barbara benson	533-69-1238	839-8461	7384 Fontana Lane	null	3.25

2.2 THE ATTRIBUTES AND TUPLES OF A RELATION STUDENT

For this relation schema, STUDENT is the name of the relation, which has seven attributes. We can specify the following previously defined domains for some of the attributes of the STUDENT relation:

Dom (Name) = Names;
Dom (SSN) = Social security numbers;
Dom(Home Phone) = Local_phone_numbers,
Dom(Office Phone) = Local_phone_numberas, and
Dom(GPA)= Grade_point_averages.

A Relation (or **relation state)**[2] r of the relation schema R(A_1, A_2,…A_n), also denoted by r(R), is a set of **n_tuples** r = (t_1, t_2 _ t_m. Each **n_tuple** t is an ordered list of n values t = < v_1, v_2,…., v_n>, where each value v_i, $1 \leq i \leq n$ is an element of dom(A_i) or is a special **null** value.

The i^{th} value v, in tuple t, which corresponds to the attribute A_i, is referred to as t[A_i]. The terms relation **intension** for the schema R and relation extension for a relation state r(R) are also commonly used.

Figure shows an example of a STUDENT relation, which corresponds to the STUDENT schema specified above. Each tuple in the relation represents a particular student entity.

We display the relation as a table, where each tuple is shown as a row and each attribute corresponds to a column header indicating a role or interpretation of the values in that column. Null values represent attributes whose values are unknown or do not exist for some individual STUDENT tuples.

The above definition of a relation can be restated as follows. A relation r(R) is a **mathematical relation** of degree n on the domains dom(A_1), dom(A_2),…., dom(A_n), which is a **subset** ofthe **Cartesian product** of the domains that define R:

r (R) $\subseteq$ (dom (A_1). dom (A_2).….dom (A_n,))

The Cartesian product specifies all possible combinations of values from the underlying domains. Hence, if we denote the number of values or cardinality of a domain D by|**D**|, and assume that all domains are finite, the total number of tuples in the Cartesian product is: |dom (A_1)|* | dom(A_2) *…….* | dom (A_2) |

Product of all these combinations, relation state at a given time-the current **Relation**

state- reflects only the valid tuples that represent a particular state pf the real world. In general, as the state of the real world changes, so does the relation, by being transformed into another relation state. However, the result of adding an attribute to represent new information that was not originally stored in the relation.
It is possible for several attributes to have the same domain. The attributes indicate different roles, or interpretations, for the domain e. g., in the STUDENT relation, the same domain Local-phone – number plays the role of Home phone, referring to the "home phone of a student," and the role of Office Phone, referring to the "office phone of the student,"

2.3 KEYS

2.3.1 PRIMARY KEYS

The term primary key is used to denote t candidate key that is chosen by t database designer as principle means identifying entities within an entity set. e attribute PART-NUMBER of the PA relation can be called as a primary k Each PART tuple/row contains a distinct PART/NUMBER value, and this value may be used to distinguish that tuple from all others in the relation. Every relation will not have a single attribute primary key, but every relation 'will' have one another they must have unique identification of some kind. If the value of Primary key was null, it would be equivalent to saying that there exists some tuple that did not have any unique identification. It was not distinguishable from other touples. We enforce the NOT NULL and UNIQUE.

2.4 RELATIONAL DATABASE DESIGN

In general, the goal of database design is to generate a set of relation schemes that allow us to store information without redundancy, yet allow us to retrieve information easily.

Given a body of data to be represented in a database, how do we decide on a suitable logical structure for it. The required relations and their attributes need to be specified. One approach is to design schemes that are in an appropriate normal form.

2.4.1 NORMAL FORMS (NF)

There are various types of NFs available. A relation is said to be in a particular normal form only if it satisfies the constraint (that it contains atomic values only to be able to call it as in 1NF.) A number of normal forms have been defined. Below is drawn a set-diagram of NFs.

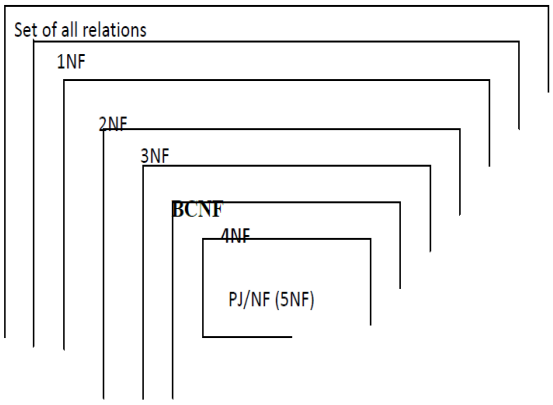


Fig: Normal Forms

1st Normal Form

Functional Dependency

Given a relation R, attribute A of R is functionally dependent on attribute B of R, if and only if, each A- value in R has associated with it precisely one 3-value in R (at any instant of time).
Referring to our PARTS relation it can be seen that PARTS = (PART_NO, PART_NAME, PART_SIZE, QTY_ON_HAND), the attributes PART_NAME, PART_SIZE, QTY_ON_HAND of it are each functionally dependent on attribute PART_No, because given a value of PART_NO, there exists precisely one corresponding value for each of PART_NAME, PARTS_SIZE etc. Using symbols it is specified as,

PARTS. PART NO → PARTS.PART NAME
PARTS.PART_NO → PARTS.PART_SIZE
PARTS.PART_NO → PARTS.QTY_ON_HAND
or more succinctly
PARTS.PART_NO → PARTS. (PART_NAME, PART_SIZE, QTY_ON_HAND)

The statement PART.PART_NO → PARTS.PART_NAME (for example) is read as 'attribute PARTS.PART_NAME is functionally dependent upon attribute PARTS.PART_NO' or equivalently, 'attribute PARTS.PART_NO functionally determine attribute PARTS. PART_NAME'.
NOTE that there is no requirement in the definition of functional dependence that a given X-Value appear in only one tuple of R.

2.4.1.1 FIRST NORMAL FORM

A relation is said to be in the 1st normal form if every attribute is functionally dependent on the key.

First normal form (1NF) is now considered to be part of the formal definition of a relation in the basic (flat) relational model; historically, it was defined to disallow multi valued attribute, 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 form 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, 1 NF disallows "relations within relations" or relations as attribute of tuples." The only attribute values permitted by 1NF are single atomic (or indivisible) values.

FULL FUNCTIONAL DEPENDENCE

Attribute B is fully functionally dependent on attribute A, if and only if, it is functionally e.g., Take a relation S,

S=(SUPLR_NO, SUPL_NAME, SUPL_STATUS, CITY)
It is supposed to maintain information on suppliers detailing,
SUPLR_NO : Unique supplier code
SUPL_NAME : Name of supplier
SUPL_STATUS : Numerical value
Indicating quality and other services provided by supplier w.r.t. city in which it is located
CITY : City where the supplier is located
In this relation S, the attribute, CITY is functionally dependent on the {SUPL_NO, SUPL_STATUS}; however, it is not fully functionally dependent on this composite attribute because, of course, it is also functionally dependent on SUPLR_NO alone. We will use this concept of full functional dependence while we discuss the normal forms on next page.

2.4.1.2 SECOND NORMAL FORM (2NF)

To be in 2NF, a table must be in first normal form and no attributes of the table should be functionally dependent on only one part of a concatenated primary key. In other words, a relation R is in second normal form (2NF) if and only if it is 1NF and every non key attribute is fully dependent on the primary key.
e. g. let`s consider ORDER_ITEM relation,
ORDER_ITEM (ORD_NO, ITEM_NAME, QTY, ITEM_PRICE)
This relation lists the items contained within various orders, the quantity of each associated with the order and their unit prices.
The key is formed as a combination of (ORD_NO + ITEM_NAME), the table is in 1NF. It is not in 2NF, because ITEM_PRICE is not functionally dependent on the ORD_NO component of the key. It is dependent only on the ITEM_NAME component. QTY is dependent on both parts of the key. To achieve 2NF form of this relation, we decompose the ORDER_ITEM relation in,

ORDERJTEM (ORD_NO, ITEM_NAME, QTY)
ITEM (ITEM_NAME, ITEM_PRICE)
As we can see in a new 2NF set of relations, it is now possible to add a new item details irrespective of whether an order is placed on it or not. If unit price of an item, ITEM_PRICE changes for a particular item then, only ITEM relation needs to be undated, effectively making less number of price updates per occurrence of an ITEM_DETAIL.

2.4.1.3 THIRD NORMAL FORM (3 NF) TRANSITIVE DEPENDENCE

If attribute A depends on B and B depends on C, due to which A depends on C, then A is said to be transitively dependent on C.
Transitive dependency causes problems in updating.

Third Normal Form (3NF)

To be in 3NF, a table must be in 2NF and no attribute of the table should be transitively functionally dependent on the primary key.
Let us consider a relation, BORROW, related to library management,
BORROW= (BOOK_NO, PERSON_NO, DURATION, LIBRARIAN_NO, LIBRARIAN_GRADE)
This relation lists books issued from various local libraries, for which duration the book was issued, the staff member of the librarian who signed out the book and grade of that librarian. To identify the issued books records/tuples, the key used is (BOOKJ_NO + PERSON_NO). Above relation is in 2NF but not in 3NF. This is because LIBRARIAN_GRADE is only transitively dependent on LIBRARIAN_NO and not dependent on (BOOK_NO+PERSON_NO).
It is functionally dependent on identity of librarian. So, we decompose the relation as,
BORROW =(BOOK_NO, PERSON_NO, DURATION, LIBRARIAN_NO)
LIBRARIAN=(LIBRARIAN_NO,IBRARIAN_GRADE)

As can be seen, it is possible to specify the grade of a librarian just by knowing LIBRARIAN_NO and irrespective of whether the librarian has issued any book or not.

Normal Form	Test	Remedy (Normalization)
First (INF)	Relation should have no non-atomic attributes or nested relations.	Form new relations for each non-atomic attribute or nested Relation
Second(2NF)	For relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each particular key with its dependent attributes(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
Third (3NF)	Relation should not have a non key attribute functionally determined by of non-key attributes). That is, there should be no transitive dependency of a non-key attribute on the primary key	Decompose and set up a relation that includes the non-key attribute(s) that functionally determine(s) other non key attribute(s).

Table: Summary of Normal Forms Based on Primary Keys and Corresponding Normalization

2.5 NETWORK DATA MODELING CONCEPTS

There two basic data structures in the network model: records and sets.

Records, Record Types, and Data Items

Data is stored in **records**; each record consists of a group of related data values. Records are classified into **record types,** where each record type describes the structure of a group of records that store

the same type of information. We give each record type a name, and we also give a name and format (data type) for each **data item** (or attribute) in the record type figure 4 shows a record **type STUDENT** with data items **NAME, SSN, ADDRESS, MAJORDEPT** & **BIRTHDATE.**

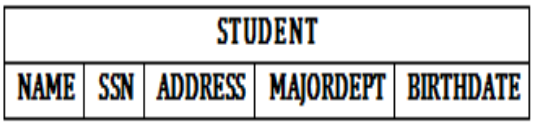
We can declare a virtual data item (or derived attribute) AGE for the record type shown in Figure 4 and write a procedure to calculate the value of AGE from the value of the actual data item **BIRTHDATE** in each record. A typical database application has numerous types–from a few to a few hundred. To represent relationships between records, the networks model provides the modeling construct called set type which we discuss next.

2.5.1 SET TYPES AND THEIR BASIC PROPERTIES

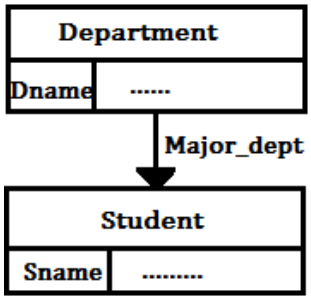
A set type is a description of a 1: N relationship between two record types. Figure 5 shows how we represent a set type diagrammatically as an arrow.

This type of diagrammatic representation is called a **Bachman diagram.** Each set type definition consists of three basic elements:-

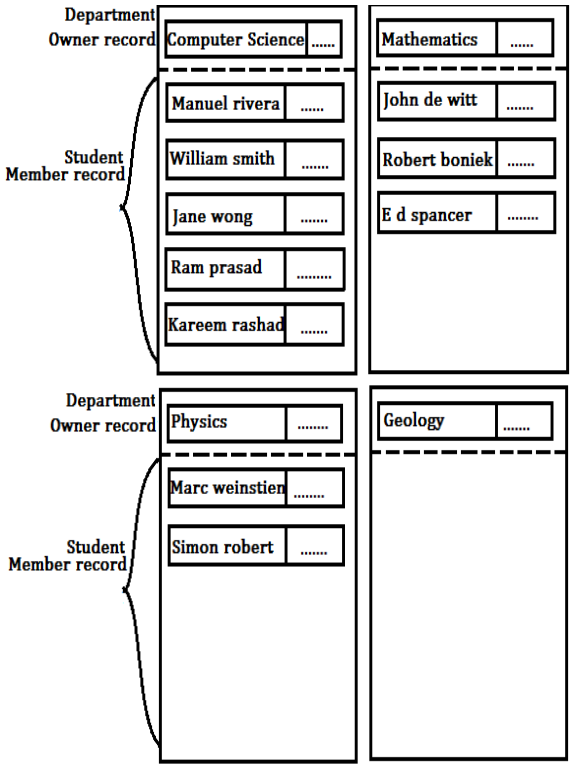
- A name for the set type.
- An owner record type.
- A member record type.



Data item name



NAME
SSN
ADDRESS
MAJORMDEPT
BIRTHDATE
Format
CHARACTER 30
CHARACTER 9
CHARACTER 40
CHARACTER 10
CHARACTER



The set type in Figure is called MAJOR_DEPT, DEPARTMENT is the owner record type, and STUDENT is the member record type. This represents the 1: N relationship between academic departments and students majoring in those departments. In the database itself, there will be many set occurrences (or set instances) corresponding to a set type. Each instance relates one record from the owner record type DEPARTMENT record in our example to the set of records from the member record type related to it the set of STUDENT records for students who major in that department. Hence, each set occurrence is composed of

- One owner record from the owner record type.

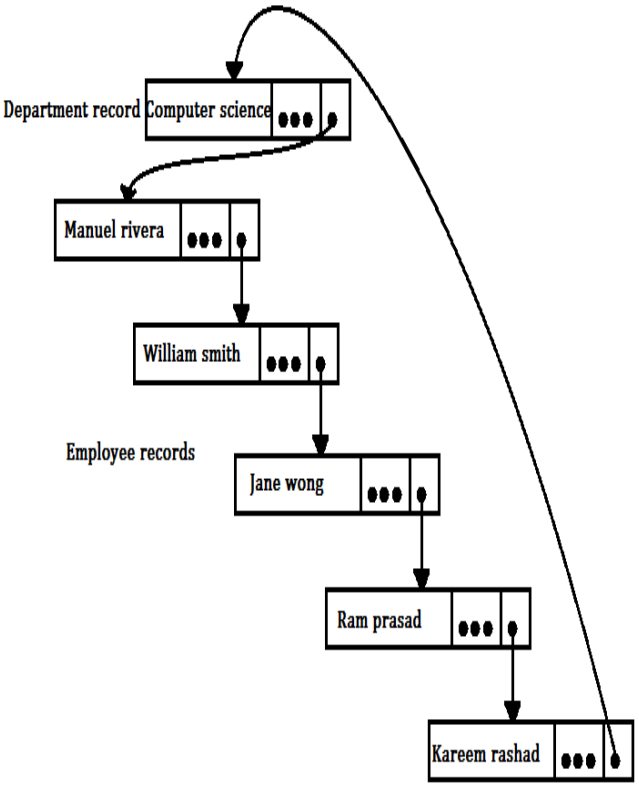
- A number of related member records (zero or more) from the member record type. A record from the member record type cannot exist in more than one set occurrence of a particular set type. This maintains the constraint that a set type represents a 1:N relationship. In our example a STUDENT record can be related to at most one major DEPARTMENT and hence is a member of at most one set occurrence of the MAJOR DEPT set type. A set occurrence can be identified either by the owner record or by any of the member records. Figure 6 shows four set occurrences (instances) of the MAJDR-DEPT set type.

Notice that each instance must have one owner record but can have any number of member records (zero or more). Hence, we usually refer to a set instance by its owner record. The set instances in Figure 6 can be referred to as the `Computer Science`, `Mathematics`, `Physics`, and `Geology` sets. It is customary to use a different representation of a set instance (Figure 7) where the records of the set instance are shown linked together by pointers, which corresponds to a commonly used technique for implementing sets.

In the network model, a set instance is not identical to the concept of a set in mathematics.

There are two principal differences:-

- The set instance has one distinguished element the owner record_ whereas in a mathematical set there is no such distinction among the elements of a set.
- In the network model, the member records of a set instance are ordered, whereas order of elements is immaterial in a mathematical set. Hence, we can refer to the first, second, i^{th}, and last member records in a set instance. Figure 7 shows an alternate "linked" representation of an instance of the set **MAJOR_DEPT.** In figure 7 the record of `Manual Riverva` is the first **STUDENT**(member) record in the 'computer Science' set, and that of 'Kareem Rashad' is the last member record. The set of the network model is sometimes referred to as an **owner-coupled set or co-set,** to distinguish it from a mathematical set



3 TRANSACTIONS

3.1 TRANSACTIONS:

Collections of operations that form a single logical unit of work are called transactions.

3.1.1 TRANSACTION CONCEPT

A transaction is a unit of program execution that accesses and possibly updates various data items. To ensure integrity of the data, we require that the database system maintains the following properties of the transactions:
If the database is consistent before an execution of the transaction, the database remains consistent after the execution of the transaction. Ensuring consistency for an individual transaction is the responsibility of the application programmer who codes the transaction.
Ensuring atomicity is the responsibility of the database system itself; specifically, it is handled by a component called the transaction-management component.
The durability property guarantees that, once a transaction completes successfully, all the updates that it carried out on the database persist, even if there is a system failure after the transaction completes execution.
Ensuring durability is the responsibility of a component of the database system called the recovery management component.
If several transactions are executed concurrently, their operations may interleave in some undesirable way, resulting in an inconsistent state.
The isolation property of a transaction ensures that the concurrent execution of transactions results in a system state that is equivalent to a state that could have been obtained had these transactions executed one at a time in some order. Ensuring the isolation property is the responsibility of a component of the database system called the concurrency-control component.

3.1.2 TRANSACTION STATE

A transaction may not always complete its execution successfully such a transaction is termed aborted. Any changes that the aborted transaction made to the database must be undone.
Once a transaction has committed, we cannot undo its effects by aborting it. The only way to undo the effects of a committed transaction is to execute a compensating transaction.

Active, the initial state; the transaction stays in this state while it is executing
Partially committed, after the final statement has been executed
Failed, after the discovery that normal execution can no longer proceed
Aborted, after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction
Committed, after successful completion

A transaction is said to have terminated if has either committed or aborted, since the actual output may still be temporarily residing in main memory.

3.1.3 IMPLEMENTATION OF ATOMICITY AND DURABILITY

The recovery-management component of a database system implements the support for atomicity and durability. We first consider a simple, but extremely inefficient, scheme. This scheme assumes that only one transaction is active at a time, and is based on making copies of the database, called shadow copies. The scheme assumes that the database is simply a file on disk. A

pointer called db-pointer is maintained on disk; it points to the current copy of the database. In the shadow-database scheme, a transaction that wants to update the database first creates a complete Copy of the database. All updates are done on the new database copy, leaving the original copy, called the shadow copy, untouched.
If at any point the transaction has to be aborted, the new copy is merely deleted. The old copy of the database has not been affected. On UNIX systems, the flush command is used of this purpose.
The transaction is said to have been committed at the point where the updated db-pointer is written to disk. We can abort the transaction by deleting simply the new copy of the database.

3.1.4 DISADVANTAGE

Since executing a single transaction requires copying the entire database.
Furthermore, the implementation does not allow transactions to execute concurrently with one another.

3.1.5 CONCURRENT EXECUTIONS

Ensuring consistency in spite concurrent execution of transactions requires extra work; it is far easier to insist that transactions run serially-that is, one at a time, each starting only after the previous one has completed. There are two good reasons for allowing concurrency. There is an increase in the throughput of the system – that is, in the number of transactions that can be executed in a given amount of time.
The processor and disk utilization also increase. Concurrent execution reduces the average response time: the average time for a transaction to be completed after it has been submitted. The database system must control the interaction among the concurrent transactions to prevent them from destroying the consistency of the database. It does so through a variety of mechanisms called concurrency-control schemes. Suppose that the two transactions are executed one at a time in the order T_1 followed by T_2.
The execution sequences just described are called schedules. They represent the chronological order in which instructions are executed in the system.
These schedules are serial. Each serial schedule consists of a sequence of instructions from various transactions, where the instructions belonging to one single transaction appear together in that schedule. Thus, for a set of n transactions, there exist n! different valid serial schedules.
With multiple transactions, the CPU time is shared among all the transactions. In general, it is not possible to predict exactly how many instructions of a transaction will be before the CPU switches to another transaction. Thus, the number of possible schedules for a set of n transactions is much larger than n!

3.2 SERILIZABLE SCHEDULE

If a concurrent schedule is equivalent to any of the serial schedule it is said to be a serializable schedule and this property is known as serializability.
It is the job of database system to ensure that any schedule that gets executed will leave the database in a consistent state. Any schedule that executed has the same effect as a schedule that could have occurred without any concurrent execution. The schedule should, in some sense, be equivalent to a serial schedule.

3.2.1 SERIALIZABILITY

The database system must control concurrent execution of transactions, to ensure that the database state remains consistent.

3.2.2 CONFLICT SERIALIZABILITY

If I_i and I_j refer to the same data item Q, then the order of the two steps may matter.

I_i = **read** (Q), I_j = **read** (Q).
Since the result of only the later of the two write instruction is preserved.
However, the value obtained by the next **read** (Q) instruction of S is affected. If there is no other **write** (Q) instruction after I_i and I_j in S, then the order I_i and I_j directly affects the final value of Q in the database state that results from schedule S.
We say that I_i and I_j conflict if they are operations by different transaction on the same data item, and at least one of these instructions is a **write** operation.
Do not conflict, we can swap the order of I_i and I_j to produce a new schedule S'. Schedules 3 and 5 both produce the same final system state.
No conflicting instructions as follows:
If a schedule S can be transformed into a schedule S' by a series of swaps of no conflicting instructions, we say that S and S' are conflict equivalent.
The concept of conflict equivalence leads to the concept of conflict serializability. We say that a schedule S is conflict serializable if it is conflict equivalent to a serial schedule.
This schedule is not conflict serializable, since it is not equivalent to either the serial schedule $<T_3,T_4>$ or the serial schedule $<T_4,T_3>$.
The **write** (B) instruction of T_5 conflicts with the **read** (B) instruction of T_1.
For the system to determine that schedule 8 produces the same outcome as the serial schedule $<T_1,T_5>$, it must analyze the computation performed by T_1 and T_5, rather than just the **read** and **write** operations.

3.2.3 VIEW SERIALIZABILITY

The schedules S and S' are said to be view equivalent if the following three conditions are met: Conditions 1 and 2 ensure that each transaction reads the same values in both schedules and, therefore, performs the same computation.

Returning to our previous examples, we note that schedule 1 is not view equivalent to schedule 2, since, in schedule 1, the value of account A read by transaction T_2 was produced by T_1, whereas this case does not hold in schedule 2.
A schedule S is view serializable if it is view equivalent to a serial schedule
Every conflict – serializable schedule is view serializable, but there are view-Inserializable schedules that are not conflict serializable. Since every pair of consecutive instructions conflicts, blind writes appear in any view – serializable schedule that is not conflict serializable.

3.3 RECOVERABILITY

In a system that allows concurrent execution, it is necessary also to ensure that any transaction T_j that is dependent on T_i is also aborted.

3.4 RECOVERABLE SCHEDULES

Most database system requires that all schedules be recoverable. A recoverable schedule is one where, for each pair of transactions T_j and $T_j$such that T_j reads a data items previously written by T_i, the commit operation of T_i appears before the commit operation of T_j.

3.5 CASCADE LESS SCHEDULES

A single transaction failure leads to a series of transaction rollbacks, is called cascading rollback. Cascading rollback is undesirable, since it leads to the undoing of a significant amount of work. It is desirable to restrict the schedules to those where cascading rollbacks cannot occur. Such schedules are called cascade less schedules. T_j reads a data item previously written by T_i, the commit operation of T_i appears before the read operation of T_j. It is easy to verify that every cascade less schedule is also recoverable.

3.6 IMPLEMENTATION OF ISOLATION

Specifically, schedules that are conflict or view serializable and cascade less satisfy these requirements. The goal of concurrency–control schemes is to provide a high degree of concurrency, while ensuring that all schedules that can be generated are conflict or view serializable, and are cascade less.

3.7 TRANSACTION DEFINITION IN SQL

If a program terminates without either of these commands, the updates are either committed or rolled back.
The definition of serialize used by the standard is that a schedule must have the same effect as would a serial schedule.
The levels of consistency specified by SQL - 92 are as follows:
Allows only committed records to be read, and further requires that, between two reads of a record by a transaction, no other transaction is allowed to update the record.
Allows only committed records to be read, may have been updated by other committed transactions.` Allows even uncommitted records to be read.

3.8 TESTING FOR SERILAZABILITY

When designing concurrency control schemes, we must show that schedules generated by the scheme are serializable. To do that, we must first understand how to determine, given a particular schedule S, whether the schedule is serializable. In this section, we shall present methods for determining conflict and view serializability. We will show that there exists a simple and efficient algorithm to determine conflict serializability. However, there is no efficient algorithm for determining view serializability.
If the precedence graph for S has a cycle, then schedule S is not conflict serializable. If the graph contains no cycles, then the schedule S is conflict serializable.

Thus, to test for conflict serializability, we need to construct the precedence graph and to invoke a cycle – detection algorithm.

3.9 TESTS FOR VIEW SERIALIZABILITY

Testing for view serializability is computationally an expensive problem.
And at least one of these transactions writes.
The **write** (Q) instructions of T_3 and T_4 are called useless writes. We need to develop a scheme for deciding whether an edge needs to be inserted in a precedence graph. A labeled precedence graph. We therefore require that any schedule produced by concurrent processing of a set of transactions will have an effect equivalent to a schedule produced when these transactions are run serially in some order. A system that guarantees this property is said ensures serializability. We can test a given schedule for conflict serializability by constructing a precedence graph for the schedule; we can test for view serializability in order 2^n time by constructing a labeled precedence graph and searching over all possible distinct labeled graphs for a graph with a cycle.

4 QUEL

4.1 QUEL

QUEL is the query language in the system INGRES. QUEL is based on relational calculus. The fundamental aspect of the languages based on relational calculus is tuple variable (which is a variable that "rangers over"" some named relation).
e.g., the query "Get supplier numbers for suppliers in London" can be expressed in QUEL as:
RANGE of SX is S
RETRIEVE (SX. S#) where SX. CITY = 'LONDON'.
The tuple here is SX that ranges over relation S.

QUERY LANGUAGE

4.2 SEQUEL

In SEQUEL, the expression following WHERE can be any expression involving the attributes of the relation following FROM, arithmetic comparisons and operations. Boolean connectives (AND.K OR NOT), set operations (UNION, INTERSECT, MINUS),
set membership (X INS, where S is a set or equivalently S CONTAINS X) and the negation of set membership (X NOT IN S OR S DOES NOT CONTAIN X).
When SEQUEL applies a mapping it does not eliminate duplicates unless told to do so by the keyword UNIQUE. In SEQUEL, assignment is indicated by preceding a query by ASSIGN TO R

4.3 ASSIGN TO R:

To assign the result to relation R. Any relational algebra expression in SEQUEL can be evaluated by applying one operator at a time and thereby computing each sub expression, hence SEQUEL is complete.

4.4 SQL

SQL allows you to communicate with the database and has the following advantages:

- Efficient
- Easy to learn and use
- Functionally complete (SQL allows you to define, retrieve, and manipulate data in the tables).

SQL is a English – like language and can be used by a range of users, including those with little or no programming experience. It is a non – procedural language. It reduces the amount of time required for creating and maintaining systems. SQL statements are not case sensitive. The statements can be written on one or more lines. Keywords cannot be abbreviated or split across lines and clauses are usually placed on separate lines for readability & ease of editing.
For executing a SQL statement places a semicolon (;) at the end of the last clause.

Statement	Description
SELECT	Retrieves data from the database.
INSERT UPDATE DELELE	Enters new rows, change existing rows, and removes unwanted rows from tables in the database, respectively. Collectively known as data manipulation Language (DML)
CREATE ALTER DROP RENAME TRUNCATE	Sets up, changes, and removes data structures from tables. Collectively known as data definition Language (DDL).
COMMIT ROLLBACK SAVEPOINT	Manages the changes made by DML statements. Changes to the data can he grouped together into logical transactions.
GRANT REVOICE	Gives or removes access rights to both the Oracle database and the structures within it Collectively known as data control language (DCL).

4.5 SELECT STATEMENT

A select statement retrieves information from the database. Using a SELECT statement, you can do the following:

4.5.1 SELECTION

You can use the selection capability in SQL to choose the rows in a table that you want returned by a query. You can use various criteria to selectively restrict the rows that you see.

4.5.2 PROJECTION

You can use the projection capability in SQL to choose the columns in a table that you want returned by your query. You can choose as few or as many columns of the table as you require.

4.5.3 JOIN

You can use the join capability in SQL to bring together data that is stored in different tables by creating a link between them. In its simplest form, a SELECT statement must include the following:

- A SELECT clause, which specifies the column to be displayed
- A FROM clause, which specifies the table containing the columns listed in the SELECT clause.

SELECT [DISTINCT] {*, column [alias],……..}
FROM table

In the Syntax:

SELECT :	is a list of one or more columns
FROM	identifies which tables
DISTINCT	suppresses duplicates
*	selects all columns
Column	selects the named column
	Alias gives selected columns different headings
FROM	table specifies the table containing the columns

Select All Columns, All Rows FROM dept;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

You can display all columns of data in a table by following the SELECT keyword with an asterisk (*).

4.5.4 SELECTING SPECIFIC COLUMNS, ALL ROWS

You can use the SELECT statement to display specific columns of the table by specifying the column names, separated by commas.

4.6 DEFINING A NULL VALUE

If a row lacks the data value for a particular column, that value is said to be null, or to contain null. A null value is a value that is unavailable, unassigned, unknown, or inapplicable. A null value is not the same as zero or a space. Zero is a number, and a space is a character.

4.7 DUPLICATE ROWS

The default display of queries is all rows, including duplicate rows
To eliminate duplicate rows in the result, include DISTINCT keyword in the SELECT clause immediately after the SELECT keyword.

4.8 CHARACTERS STRINGS AND DATES

- Character strings and date values are enclosed in single quotation marks.
- Character values are case sensitive and date values are format sensitive.
- Number values are not enclosed within quotation marks.

4.9 COMPARISON OPERATORS

Comparison operators are used in conditions that compare one expression to

another. They are used in the WHERE clause in the following format:

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

4.9.1 THE BETWEEN OPERATOR

You can display rows based on range of values using the BETWEEN operator. The range that you specify contains a lower range and an upper range. Values specified with the BETWEEN operator are inclusive.

4.9.2 THE IN OPERATOR

To test for values in a specified list, use the IN operator. The IN operator can be used with any data type. If characters or dates are used in the list, they must be enclosed in single quotation marks (‘’).

4.9.3 THE LIKE OPERATOR

You can select rows that match a character pattern by using the LIKE operator. The character pattern – matching operation is referred to as a wildcard search. Two symbols can be used to construct the search string.

Symbol	Description
%	Represents any sequence of zero or more characters
-	Represents any single character

When you need to have exact match for the actual ‘%’ and ‘- ‘characters, use the ESCAPE option. This option specifies what the ESCAPE character is. If you have HEAD QUARTERS as a department name, you would search for it using the following SQL statement:

SELECT * FROM dept
WHERE dname LIKE ‘%_%’ ESCAPE ‘\’;

The ESCAPE option identifies the backslash (\) as the escape character.

4.9.4 USING THE IS NULL OPERATOR

The IS NULL operator tests for values that are null. A null value means the value is unavailable, unassigned, unknown, or inapplicable. Therefore, you cannot test with (=) because a null value cannot be equal or unequal to any value.

SELECT ename, mgr
FROM EMP
WHRER mgr IS NULL;

4.10 LOGICAL OPERATORS

A logical operator combines the result of two component conditions to produce a single result based on them or to invert the result of a single condition. Three logical operators are available in SQL.

- **AND**
- **OR**
- **NOT**

Operator	Meaning
AND	Returns TRUE if both component conditions are TRUE
OR	Returns TRUE if either component conditions are TRUE
NOT	Returns TRUE if the following conditions is FALSE

4.10.1 RULES OF PRECEDENCE

Order Evaluated	Operator
1	All comparison operators
2	NOT
3	AND
4	OR

Override rules of precedence by using parentheses.

4.10.2 THE ORDER BY CLAUSE

The ORDER BY clause can be used to sort

the rows.

SELECT **expr**
FROM **table**
[WHRER condition(s)]
[ORDER BY {column, expr}
[ASC| DESC]];

In the syntax

ORDER BY specifies the order in which the retrieved rows are displayed.
ASC orders the rows in ascending order (this is the default order).
DESC orders the rows in descending order.

i) Single-Row Functions
These functions operate on single rows only and return one result per row. There are different types of single – row functions.

- Character
- Number
- Date
- Conversion

ii)Multiple –Row Functions
These functions manipulate groups of rows to give one result per group of rows.

4.11 CHARACTER FUNCTIONS

Number Functions

LOWER	**Syntax**	**LOWER (column : expression)**
	Purpose	Converts alpha character values to lowercase
	Example	SELECT LOWER (‘ORACLE’) “Lower” FROM dual;
	Output	Lower
		_____
		Oracle
UPPER	**Syntax**	**UPPER (column : expression)**
	Purpose	Converts alpha character vales to uppercase
	Example	SELECT LOWER (‘ORACLE’) “Upper” FROM dual;
	Output	Lower
		_____
		Oracle

INITCAP	**Syntax**	**INITCAP (column : expression)**
	Purpose	Converts alpha character values to uppercase for the first character of each word, all other letters in lowercase.
	Example	SELECT INITCAP (‘ORACLE’) “Mix” FROM dual;
	Output	Mix
		_____
		Oracle
CONCAT	Syntax	CONCAT (column : expression1 column2: expression)
	Purpose	Concatenates the first character value to the second character value; equivalent to concatenation operator (||)
	Example	CONCAT (‘Good’, String’)
	Output	Good String
SUBSTR	Syntax	SUBSTR (column : expression, m, [n])
	Purpose	Returns specified characters from character values starting at character position m, n characters long (If m is negative, the count starts from the end of the character value. If n is omitted, all characters to the end of the string are returned.)
	Example	SUBSTR (‘String’, 1,3)
	Output	Str
LENGTH	Syntax	LENGTH (column : expression)
	Purpose	Returns the number of characters in value
	Example	LENGTH (‘String’)
	Output	6
INSTER	Syntax	INSTR (column : expression, m)
	Purpose	Returns the number of character in value
	Example	INSTR (‘String’, ‘r’)
	Output	3
LPAD	Syntax	LPAD (column : expression, n, ‘string’
	Purpose	Pads the character value right – justified to a total width of n character positions

	Example	LPAD (sale, 10, '*')
	Output	******5000
TRIM	Syntax	TRIM (leading : trailing: both, trim character FROM trim source)
	Purpose	Enables you to trim leading or trailing character (or both) from a character string. If trim character or trim source is a character literal, you must enclose it in single quotes.

	Example	('S' FROM 'SMITH')
	Output	MITH
LTRIM	Syntax	LTRIM (string [,'set'])
	Purpose	The function trims off unwanted characters from the left end of the string. String is the name of the column, literal string
		Set is the collection of characters you want to trim off. If no set of characters is specified, the function trims off spaces.
	Example	LTIRM ('OAORACLE', 'OA')
	Output	ORACLE
RTRIM	Syntax	RTRIM (string [, 'set'])
	Purpose	The function trims off unwanted characters from the right end of the string. String is the name of the column, literal string. Set is the collection of characters you want to trim off. If no set of characters is specified, the function trims off spaces.
	Example	RTIRM ('ORACLE.".','. " ')
	Output	ORACLE

ABS	**Syntax**	**ABS (column : expression)**
	Purpose	Returns the Absolute value of expression or column
	Example	ABS (-30)
	Output	30
CEIL	**Syntax**	**CEIL (column : expression)**

	Purpose	Returns the smallest integer that is greater than or equal to a specific value.
	Example	CEIL (1.3)
		CEIL (-2.3)
	Output	2
		-2
FLOOR	**Syntax**	**FLOOR (column : expression)**
	Purpose	FLOOR is intuitive opposite of CEIL.
	Example	FLOOR (1.3)
		FLOOR (-2.3)
	Output	01-Mar
MOD	**Syntax**	**MOD (m,n)**
	Purpose	Returns the remainder of m divided by n
	Example	MOD (-30,7)
		MOD (4.1, .3)
	Output	-2 0.2
POWER	**Syntax**	**POWER (value, exponent)**
	Purpose	Raise the value to a given exponent
	Example	POWER (3,3)
	Output	27
ROUND	**Syntax**	**ROUND (column : expression, n)**
	Purpose	Rounds the column, expression, or value to n decimal places or if n is an omitted, no decimal place. If n is negative, number to left of the decimal points are rounded.
	Example	ROUND (45.926,2)
	Output	45.93
SQRT	**Syntax**	**SQRT (column : expression)**
	Purpose	Returns the square root of the given number. The number must be positive, because square root of negative number is an imaginary number.
	Example	SQRT (64)
	Output	8
TRUNC	**Syntax**	**TRUNC (column : expression, n)**
	Purpose	Truncates the column, expression or value to n

		decimal places or if n is omitted, no decimal places. If n is negative, numbers left of the decimal point are truncated to zero.
	Example	TRUNC (45.926,2)
	Output	45.92

4.12 WORKING WITH DATES

The dates are stored in an internal numeric format, representing the century, year, month, day, hours, minutes and seconds.

4.12.1 SYSDATE

SYSDATE is a date function that returns the current date and time. You can use SYSDATE just as you would use any other column name. For example, you can display the current date by selecting SYSDATE from a table.

4.12.1.1 SELECT SYSDARE FROM DUAL; Using Arithmetic Operators with Dates:

Since the database stores dates as numbers, you can perform calculations using arithmetic operators such as addition and subtraction. You can perform the following operations:

Operation	Result	Description
Date + number	Date	Adds a number of days to a date
Date - number	Date	Subtracts a number of days from a date
Date-Date	Number of days	Subtract one date from another
Date A-number / 24	Date	Adds a number of hours to a date

4.12.1.2 Date Functions

MONTHS_ BETWEEN	**Syntax**	**MONTHS_BETWEEN (date1, date2)**
	Purpose	Finds the number of months between date1 and date2. If date 1 is later than date2, the result is positive; if date1 is earlier than date2, the result is negative. The non-integer part of the result represents a portion of the month.
	Example	MONTHS_BETWEEN ('01 – SEP-95','11-JAN-94')
	Output	19.6774194
ADD_MON THS	**Syntax**	**ADD_MONTHS (date, 'char')**
	Purpose	Adds n number of calendar months to date. The value of n must be an integer and can be negative.
	Purpose	Adds n number of calendar months to date. The value of n must be an integer and can be negative.
	Example	ADD_MONTHS ('11-JAN-94', 6)
	Output	11-Jul-94
NEXT_DA Y	**Syntax**	**NEXT_DAY (date, 'char')**
	Purpose	Finds the date of the next specified day of the week ('char') following date. The value of char may be a number representing a day of a character string.
	Example	NEXT_DAY ('01-SEP-95', 'FRIDAY')
	Output	08-Sep-95
LAST_DAY	**Syntax**	**LAST_DAY (date)**
	Purpose	Finds the date of the last day of the month that contains date.
	Example	LAST_DAY ('01-SEP-95')
	Output	30-Sep-95
ROUND	**Syntax**	**ROUND (date['fmt'])**
	Purpose	Returns date rounded to the unit specified by the format model fmt. If the format model fmt is omitted, date is rounded to the nearest day.

	Example	ROUND ('25-JUL-95','MONTH')
		ROUND ('25-JUL-95','YEAR')
	Output	01-Aug-95 01-Jan-96
TRUNC	**Syntax**	**TRUNC (date [,'fmt'])**
	Purpose	Returns date with the time portion of the day truncated to the unit specified by the format model fmt. If the format model fmt is omitted, date is truncated to the nearest day.
	Example	TRUNCATE ('25-JUL-95','MONTH') TRUNCATE ('25-JUL-95','YEAR)
	Output	01-Jul-95 01-Jan-95

Explicit Data type Conversion:

SQL provides three functions to convert a value from one data type to another.

Function	Purpose
TO_CHAR(number | date [fmt])	Converts a number or date value to a VARCHAR2 character string with format mode 'fmt'.
TO_NUMBER(char [fmt])	Converts a character string containing digits to a number in the format specification by the optional format model 'fmt'.
TO_DATE(char [fmt])	Converts a character string representing a date to a date value according to the 'fmt' specified. If 'fmt' is omitted, the format is DD-MM-YY.

4.12.1.3 DISPLAYING DATA FROM MULTIPLE TABLES

When data from more than one table in the data base is required, a join condition is used. Rows in one table can be joined to rows in another table according to common values existing in corresponding columns, that is, usually primary & foreign key columns.

Syntax:
SELECT table. Column, table2. Column FROM table 1, table2 WHERE table1.column1=table2.column2;
If the same column name appears in more than one table, the column name must be prefixed with the table name.

4.13 TYPES OF JOINS

There are two main types of join conditions:

- Equijoin
- Non – equijoin

4.13.1 Additional join methods include the following:

- Outer join
- Self join
- Set operators

4.13.1.1 Set Operators

Set operators combines results of two queries into a single result. Set operations are generally performed on two Lists obtained from distinct tables.

Operator	Description
UNION	All distinct rows selected by either query
UNION ALL	Rows selected by either query, including all duplicates.
INTERSECT	All distinct common rows selected by both queries.
MINUS	All distinct rows selected by the first query but not by the second

4.14 GROUP FUNCTIONS:

Unlike single-row functions, group functions operate on sets of rows to give one result per group. The sets may be the whole table or the table split into groups.

4.14.1 TYPES OF GROUP FUNCTIONS

Function	Description
VG ([DISTINCT : ALL]n)	Average value of n, ignoring null values
COUNT ({*:[DISTINCT: ALL]expr})	Number of rows, where naps evaluates to something other than null (Count an selected rows using*, including duplicates and rows with nulls)
MAX ([DLSTINCT: ALL]expr})	Maximum value of expr, ignoring null values
MIN ([DISTINCT: ALL]expr)	Minimum value of min, ignoring null values
SUM ([DISTNCE: ALL]n)	Sum values of n, ignoring null values
STDDSV ([DISNCT:|ALL]n)	Standard deviation of lapin-Mg null values
VARIANCE ([DISTINCT :ALL]n)	Variance of n, ignoring null values

The COUNT function has two formats:

- COUNT (*)
- COUNT (expr)

COUNT (*) returns the number of rows in a table, including duplicate rows and rows containing null values in any of the columns.

COUNT (*) returns the number of rows satisfies the condition in the WHERE clause

```
SELECT    COUNT (*)
FROM      emp
WHERE     deptno = 30;
```

COUNT (expr) returns the number of non null rows in the column identified by expr.

```
SELECT    COUNT (comm.)
FROM      emp
WHERE     deptno = 30;
```

4.15 SUB QUERIES

The inner query or the sub query returns a value that is used by the outer query or the main query.

4.15.1 TYPES OF SUBQUERIES

(i) Single – row Subquereis:
Queries that return only one row form the inner SELECT statement.

(ii) Multiple –row Subqueries:
Queries that return more than one row from the inner SELECT statement.

(iii)Multiple – column Sub queries:
Queries that return more than one column from the inner SELECT statement.

4.15.1.1 Single –Row Sub queries

A single – row sub query is one that returns one row from the inner SELECT statement. This type of sub query uses a single – row operator.

HAVING Clause with Sub queries

You can use sub queries not only in the WHERE clause, but also in the HAVING clause. The Server executes the sub query, and the results are retuned into the HAVING clause of the main query.

```
SELECT    deptno, MIN (Sal)
FROM      emp
GROUP BY  deptno
HAVING    MIN (Sal) >
SELECT    MIN (Sal)
FROM      emp
WHERE     deptno = 20);
```

4.15.1.2Multiple – Row Sub queries

Sub queries that return more than one row are called multiple – row sub queries. You use a multiple – row operator, instead of single – row operator, with a multiple – row sub query. The multiple – row operator expects one or more values.

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the sub query
All	Compare value to every value returned by the sub query

Example 5.2

Find the employees who earn the same salary as the minimum salary for departments.

```
SELECT    ename, Sal, deptno
FROM      emp
```

```
WHERE     Sal IN
(SELECT MIN (Sal)
FROM      emp
GROUP BY  deptno);
```

The inner query is executed first,

4.15.1.3 Multiple – Column Sub queries

If you want to compare two or more columns, you must write a compound WHERE clause using logical operators. Multiple –column sub queries enable you to combine duplicate WHERE conditions into a single WHERE clause.

Syntax

```
SELECT    column, column,
FROM      table
WHERE     (column, column...) IN
          (SELECT column, column,
          FROM table
          WHERE condition);
SELECT    empno, mgr, deptno
FROM      emp
WHERE     (mgr,deptno) IN
          (SELECT mgr, deptno
          FROM emp
          WHERE ename =' MARTIN')
AND       ename<> 'MARTIN';
```

4.15.1.4 CORRELATED SUBQURIES

The Server performs a correlated sub query when the sub query references a column from a table referred to in the parent statement. A correlated sub query is evaluated once for each row processed by the parent statement. The parent statement can be a SELECT, UPDATE or DELETE statement. The Server performs a correlated sub query when the sub query references a column from table in the parent query.

Syntax

```
SELECT    column1, column2, …….
FROM      table1 alias
WHERE     column1, operator
          (SELECT column1,
     column2……..
          FROM table2
          WHERE expr1 =
     alias.expr2)
```

Example 5.3

```
SELECT    ename, sal, deptno
FROM      emp
WHERE     deptno = e.deptno);
```

4.16 Data Definition Language (DDL)

4.16.1 DATABASE OBJECTS

A database can contain multiple data structures. Each structure should be outlined in the database design so that it can be created during the build stage of database development.

4.16.2 THE CREATE TABLE STATEMENT

Create tables to store data by executing the SQL, 'CREATE TABLE' statement. This statement is one of the data definition language (DDL) statements.

Syntax

```
CREATETABLE    [schema.] table
(column    data    type    [DEFAULT
     expr][,…..]);
```

In the syntax

Schema Is the same as the owner's name
Table Is the name of the table
DEFAULT expr Specifies a default value if a value is omitted in the INSERT Statement
Column Is the name of the column
Data type Is the column's datatype and length

4.16.3 DATATYPES

Data type	Description
VARCHAR2 (size)	Variable-length character data (A maximum size must be specified. Default and minimum size is 1. Maximum size is 4000)
CHAR (size)	Fixed - length character data of length size bytes (Default and minimum size Is 1. Maximum size is 2000)
NUMBER(p, s)	Number having precision p and scales.(The precision is the total

	number of decimal digits and the scale is the number of digits to the right of the decimal point. The precision can range from 1 to 38 and the scale can range from 84 to 127.)
DATE	Date and time values between January 1, 4712 B.C. and December 31, 9999 A.D.
LONG	Variable-length character data up to 2 gigabytes.
CLOB	Single-byte character data up to 4 gigabytes.
RAW (size)	Paw binary dam of length of size (Size must be specified Maximum size 2000).
LONG RAW	Raw binary data of variable length up to 2 gigabytes.
BLOB	Binary dam up to 4 gigabytes.
BBILE	Binary data stored in an external file; up 4 gigabytes.

4.16.4 CREATING A TABLE BY USING SUBQUERY

A second method to create a table is to apply the as sub query clause to both create the table and insert rows returned from the sub query.

```
CREATE      TABLE      table
            [(Column, column….)]
As sub query;
CREATE      TABLE      dept30
AS
      SELECT      empno,      ename,
sal*12ANNSAL,
       hiredate
            FROM       emp
            WHERE      deptno=30;
```

4.16.5 THE ALTER TABLE STATEMENT

After you create your tables, you may need to change the table structures because you omitted a column or your column definition need to be changed. You can do this by using the ALTER TABLE statement.

You can use the ALTER TABLE statement to:

- Add a new column.
- Modify an existing column.
- Define a default value for the new column.

4.16.5.1 Add a Column

You can add columns to a table by using the ALTER TABLE statement with the ADD clause.

```
ALTER   TABLE
ADD     (column data type [DEFAULT
expr]
        [, column data type]…..);
```

4.16.5.2 Modify a Column

You can modify existing columns in a table by using the ALTER TABLE statement with the MODIFY clause. Column modification can include changes to a column's data type, size and default value.

```
ALTER       TABLE table
MODIFY      (column          datatype
            [DEFAULT expr]
            [, column datatype]….);
ALTER       TABLE       dept30
MODIFY      (enameVARCHAR2(15) );
Table altered
```

4.16.5.3 Dropping a Column

You can drop a column from a table by using the ALTER TABLE statement with the DROP COLUMN clause.

```
ALTER       TABLE       dep30
DROP        COLUMN      job;
Table       altered.
```

Guidelines

- The column may or may not contain data
- Only one column can be dropped at a time.
- The table must have at least one column remaining in it after it is altered
- Once a column is dropped, it cannot be recovered

4.16.5.4 DROPPING A TABLE

The DROP TABLE statement removes the definition of a table. When you drop a table, the database looses all the data in the table and all the indexes associated with it.

Syntax

DROP TABLE table;

where table is the name of the table

DROP TABLE dept30;

Table dropped

4.16.5.5 TRUNCATING TABLE

Another DDL statement is the TRUNCATE TABLE statement, which is used to remove all rows from a table and to release the storage space used by that table. When using the TRUNCATE TABLE statement, you cannot rollback row removal.

Syntax

TRUNCATE TABLE;

4.16.6 INCLUDIG CONSTRAINTS

An integrity constraint can be thought of as a way to define a business rule for a column or a table. Integrity constraints are defined with a table and are stored as part of the table's definition in the data dictionary.

4.16.6.1 Data Integrity Constraints

Constraint	Description
NOT NULL	Specification that this column may not contain a null value
UNIQUE	Specifies a column or combination of column whose value must be unique for all rows in table
PRIMARY KEY	Uniquely identifies each row of the table
FOREIGGN KEY	Establishes and enforces a foreign key relationship between the column and a column of the referenced table.
CHECK	Specifies a condition that must be true.

4.16.6.2 Defining Constraints

The integrity constraints can be defined using the CONSTRAINT clause. The CONSTRAINT clause can appear in CREATE TABLE and ALTER TABLE commands. This clause allows the user to define UNIQUE, PRIMARY KEY, NOT NULL, FOREIGN KEY and CHECK integrity constraints. The CREATE TABLE command is used when the constraints are to be defined at the time of table creation. The ALTER TABLE command is used when the table already exists and constraints are to be defined.

CREATE TABLE [schema.] table
(columndatatype [DEFAULT expr]
[column_constraint],……..
[table_constraint] [,....]);

4.16.6.3 Disabling Constraints

You can disable a constraint without dropping it or re-creating it by using the ALTER TABLE statement with the DISABLE clause.

Syntax

ALTER TABLE table
DISABLE CONSTRAINT constraint [CASCADE];

ALTER TABLE emp
DISABLE CONSTRIANT emp_empno_pk CASCADE;

4.17 VIEWS

A view is a logical table based on a table based on a table or another view. A view contains no data of its own but is like a window through which data from tables can be viewed or changed. The tables on which a view is based are called base tables. The view is stored as a SELECT statement in the data dictionary.

4.17.1 Advantages of Views

- Views restrict access to the data because the view can display selective columns from the table.
- Views allow users to make simple queries to retrieve the results from

complicated queries. For example, views allow users to query information from multiple tables without knowing how to write a joint statement.

- Views provide data independence for adhoc users and application programs. One view can be used to retrieve data from several tables.
- Views provide groups of users access to data according to their particular criteria.
- Views avoid data redundancy.
- Views provide data security.

4.17.2 Creating Views

You can create a view by embedding a subqery within the CREATE VIEW statement.

```
CREATE [OR REPLACE] VIEW view
 [ (alias [, alisas]....)]
AS subquery;
CREATE VIEW empvu10
AS SELECT empno, ename, job
FROM emp
WHERE deptno = 10;
View created.
```

4.17.3 Removing a View

You use the DROP VIEW statement to remove a view.

```
DROP VIEW;
DROP VIEW empvu10;
     View dropped.
```

4.17.4 Inline Views

- An inline view is a sub query with an alias (correlation name) that you can use within a SQL statement.
- An inline view is similar to using a named subquery in the FROM clause of the main query.
- An inline view is not a schema object.

An inline view in the FROM clause of a SELECT statement defines a data source for the SELECT statement.

```
SELECT a.ename, a.sal, a.deptno,
b.maxsal
FROM emp a, (SELECT deptno, max(sal)
maxsal
     FROM emp
     GROUP BY deptno) b
WHERE a.deptno = b.deptno
AND a.sal<b.maxsal;
```

4.18 DATA MANIPULATION LANGUAGE (DML)

Data Manipulation Language (DML) is a core part of SQL. When you want to add, update or delete data in the database, you execute a DML statement. A collection of DML statements that form a logical unit of work is called transaction.

4.18.1 DML Statements:

Statement	Purpose
Insert	Add rows to table
Update	Change the values in table
Delete	Remove rows from a table

4.18.2 The INSERT Statement

You can add new rows to a table by issuing The INSERT statement.

```
INSERT INTO table [(column
[,column....])]
VALUES (value [, value.....]);
INSERT INTO dept (deptno, dname, loc)
VALUES (50,'DEVELOPMENT','DETROIT');
```

4.18.3 The UPDATE Statement

You can modify existing rows by using the UPDATE statement.

```
Syntax
UPDATE table
SET column=value
[, column=value,...]
[WHERE condition];
UPDATE emp
SET deptno = 20
WHERE empno = 7782;
```

4.18.4 The DELETE Statement

You can remove existing from a table by using the DELETE statement.

Syntax

```
DELETE    [FROM] table
[WHERE    condition];
DELETE    FROM department
WHERE     dname= 'DEVELOPMENT';
```

4.18.4.1 QBE QUERY

QBE is specifically designed for use with a visual display terminal QBE has a two-dimensional syntax since operations are specified in tabular form.
The basic idea is that the user formulates the query by entering an example of a possible answer in the appropriate place in an empty table. For example, consider the query 'Get supplier numbers for suppliers in Paris'. First by pressing a certain function key on the terminal, the user can have a blank "skeleton" table displaced on the screen. Then the user enters S as the table-name gets QBE to respond by filling in the column names. The query is expressed as

	S#	SNAME	STATUS	CITY
	P.S7			PARIS

The "P" stands for "print", indicating the target of the query. S7 is an "example element" i. e. , an example of a possible answer to the query(indicated by underlining). PARIS is a "constant element" QBE automatically eliminates redundant duplicates from a query result. If suppress each elimination the user can specify the key word "ALL".

4.18.4.2 SQL (STRUCTURED QUERY LANGUAGE)

This language consists of commands; each command has an operational part and condition part. Operation is executed through a search in all relations defined in relational database. Required solution (list of data) is returned by the command. The data listed in solution satisfies all conditions given in that command.
The name SQL is derived from Structured Query Language. Originally, SQL was called SEQUEL (for Structured English QUERY language) and was designed and implemented at IBM Research, as the interface for an experimental relational database system called SYS-TEM R. SQL is now the standard language for commercial relational DBMS. A joint effort by ANSI (the American National Standards Institute) and ISO (the International Standards Organization) has led to a standard version of SQL (ANSI 1986), called SQL-86 or SQL1. A revised and much expanded standard called SQL 2 (also referred to as SQL-92) has extended SQL with object-oriented and other recent database concepts.
SQL is a comprehensive database language; it has statements for data definition, query, and update. Hence, it is both a DDL and a DML. In addition, it has facilities for defining views on the database, for specifying security and authorization, for defining integrity constrains and for specifying transaction controls. It also has rules for embedding SQL statements into a general-purpose programming language such as C or PASCAL.

4.18.4.3 SQL commands SQL

provides commands for a variety of tasks including

1) Querying data
2) Inserting, updating and deleting rows in a table
3) Creating, altering and dropping object
4) Controlling access to the database and its objects
5) Guaranteeing database consistency

4.18.5 Types of SQL Statements SQL

statements can be divided into three major categories,

1) Data Manipulation Language
2) Data Definition Language, and
3) Transaction Control language.

1) Data Manipulation language:

These statements consist of queries that retrieve data from tables in a database and

statements that change in the database. The statements under this category are:
SELECT
INSERT
UPDATE
DELETE
LOCK TABLE etc.

2) Data Definition Language

These statements define the structure of the database. DBL consists of those statements that create, alter and drop database objects and statements that grant and revoke privileges and roles to user of the database. The statements under this category are:
CREATE
ALTER
DROP
GRANT
REVOKE etc.

3) Transaction Control Language

Transaction control commands manage changes made by data manipulation language commands. These commands are:
COMMIT
ROLLBACK etc.

4.18.6 DATA DEFINITION LANGUAGE
Object Naming Rules

Some rules for names of objects and their parts are:

- Names must be from 1 to 30 bytes long
- Names are not case sensitive
- A name must begin with an alphabet
- Names can only contain alphanumeric characters from database
- A name cannot be an Oracle reserved word.

4.18.7 DATA Types

1. VARCHAR2 (Size)
It is variable length character string having maximum length 'Size' bytes. Maximum size is 2000.

2. CHAR (Size)
It specifies a fixed length character string. Maximum size is 255. Default size is 1 byte. If you insert a value that is shorter than the column length, ORACLE black-pads the value to column length. Trying to insert a value too large for column returns an error.

3. VARCHAR (Size)
The VARCHAR data type is currently synonymous with VARCHAR2 data type. ORACLE Corporation recommends use of VARCHAR2 rather than VARCHAR as meaning of VARCHAR may change in future versions.

4. NUMBER (P, S)
It is used to store fixed or floating point numbers ranging from $1.0^{*}\ 10^{-130}$ to $9.99^{*}\ 10^{125}$ P-Precision or total number of digits range 1 to 38.

5. LONG
It stores variable length character strings containing upto 2 gigabytes. **LONG** columns can be referenced in SQL statements in these places:

- Select lists
- SET clause of UPDATE statement
- VALUES clauses of INSERT statement

6. DATE
It is used to date and time information. Default format in DD-MM-YY. Default format can be changed by the NLS-DATE-FORMAT parameter in int. ora file

7. ROW ID
Each row in the database has an address of the type **block.row.file,** where,
Block: is a hexadecimal string identifying the data block of the data file containing the row length of the string depends on your operating system.
Row: A four-digit hexadecimal string identifying the row in the data block. The first row in the block has the number 0.

File: Hexadecimal string identifying the database file containing the row. Length depends on operating system. First data file has number 1.

Example:
0000000F.0000.0002 identifies the database file containing the row. Length depends on operating system. First data file has number 1.

4.18.8 Creating Database objects : Create Command

Create statement used to create database objects. e. g. Tables, Views, Snonyms etc. The syntax of the CREATE TABLE statement is as follows:

CREATE TABLE Table-name
(Column-name 1 data-type [column constraint],
(Column-name 2 data-type [column constraint],
- - -
- - -
Column-name n data-type[column constraint],
[Table-constraints]);

The following statement creates an EMPLOYEE-DETAILS table:

CREATE TABLE Employee-details
(Emp_No Number (6) PRIMARY KEY,
Emp_Name Char (30) NOT NULL
DEFT Char (20),
SALARY Number
Constraint DEPT_CHECK
CHECK (DEFT IN ('ACCOUNT', 'PERSONNEL1',
'ADMIN'))
);

4.18.9 Altering the Definition of Table: Alter Command

ALTER TABLE command is used to alter the definition of a table in the database.
The syntax of the command is:
ALTER TABLE name
ADD column namedata_type [column constraint],
[MODIFYcolumn-namedata_type [column_constraint]];

Example:
Alter Table Employee-details add age number;

4.18.10 VIEWS

A view is a logical table that allows you to access data from other tables and views. A view contains no data itself. The tables upon which a view is based are called base tables.

Views are used to:
—Provide additional level of table security, by restricting access to a predetermined Set of rows/columns of a base table.
—To hide data complexity, for example, a view based on the join of several tables, actsas a single table.

Syntax:
CREATE VIEW View-name AS sub query [WITH CHECK OPTION].

Example:
—CREATE VIEW dept 20
AS SELECT ename, Sal FROM BMP
WHERE DEPT NO = 20
—CREATE VIEW clerk (id-number, person, department, position)
AS SELECT EMPNO, ENAME, DEPTNO, JOB FROM EMP WHERE JOB = 'CLERK' WITH CHECK OPTION
In this example, the column names in the view have been renamed.

WITH CHECK OPTION:
Specific that inserts and updates performed through the view must result in rows for which the **WHERE** clause of the view is true. You cannot perform insert, updates or deletes on a view that is based on a join of multiple tables.

4.18.11 DATA MANIPULATION LANGUAGE

Insert Command
–INSERT command is used to add rows to a table.

INSERT INTO TABLE-NAME VALUES (data value 1, data value 2);
The number and sequence of data values should match that of columns in the table. If the number of data values is less, then specify the column names into which data is being entered as illustrated.
INSERT INTO TABLE-NAME(column1, column3)
VALUES (data value 1, data value 3)
To insert null values, NULL may be used
INSERT INTO EMP VALUES (1001, `Sharma` NULL. NULL 3000, NULL);

4.19 QUERIES

Query has two components, a SELECT clause and a FORM clause.
The SELECT clause lists names of columns containing the required data and the FROM clause specifies table in which these columns are located. The SELECT clause can also clause specifies operation to be performed on the data and displays the result of these operations. The result of the query is displayed in a table form and is sometimes called the result table. The rows in the result represent the data that meet the conditions. If no data qualifies, zero rows are selected.

4.19.1 NARROWING THE QUERY: THE WHERE CLAUSE

The WHERE clause narrows the scope of the query by focusing on selected rows
Example:
SELECTNAME, TRAINER
FROM GUEST-ROSTER
WHERETRAINER = "TODD",

NAME	TRAINER
SEAN PENCIL	TODD
DON JACKSON	TODD
DON JACKSON	TODD
DIANCA JOGGER	TODD

4.19.2 RELATIONAL PREDICATES

TRAINER="IODD" is a relational predicate. The `=` is a simple relational operator.
There are nine simple relational operators:

=	is equal to
I	is not equal to
<>	is not equal to
>	is greater than
!>	is not greater than
<	is less than
!<	is not less than
>=	is greater than or equal to
<=	is less than or equal to

4.19.3 OTHER RELATION PREDICTIONS

There are four relational operators that can be used to form relational Predicates.
These relational operators are:

BETWEEN.....
AND
IS NULL
IN
EXIST

Example-
Consider the following relational database corresponding to the company scheme

Employees	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	12345	1965-01-19	731, Fondrem, Huston, TX	M	30000	33344555	5
	Franklin	T	Wong	33344	1955-12-08	638, Vos, Huston, TX	M	40000	88866555	5
	Alicia	J	Zelaya	99988	1968-07-19	3321, Castle spring, TX	F	25000	967654321	4
	Jennifer	S	Wallace	96765	1941-06-20	291, Ballerie, TX	F	43000	88866555	4
	Ramesh	K	Narayan	66688	1962-09-15	975,Fire oak Humble, TX	M	38000	334455545	5
	Joyce	A	English	4534	1972-07-29	908,Fire oak Humble, TX	F	25000	63333444	5
	Ahmad	V	Jabber	9679	1969-02-29	975,Dallas Humble, TX	M	25000	956564966	5
	James	E	Borg	88866	1937-11-10	450,Dallas Humble, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	HOUSTON
	2	STAFFORD
	3	BALLERIE
	4	SUGARLAND
	5	HOUSTON

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-06-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-10
	Headquarters	1	888665555	1981-06-10

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	B DATE	RELATIONSHIP

4.20 ONE POSSIBLE DATABASE STATE CORRESPONDING TO THE COMPANY SCHEME

Example:
Make a query for database given in fig. retrieve the birthdates and address of the employee(s) whose name is `John B. Smith`.

PROJECT	P NAME	P NUMBER	P LOCATION	D NUM
	Product X	1	Bellalre	5
	Product Y	2	Sugarland	5
	Product Z	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	New benefits	30	Stafford	4

Works_on	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666994444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

333445555 Headquarters

SELECT **B/DATE, ADDRESS**
FROM **EMPLOYEE**
WHERE **FNAME** =`John` AND MINT = B AND LNAME = `Smith`;

Which gives the following results

BDATE **ADDRESS**
1965-01-09 **731 FONDREN, HOUSTON, TX**

Example:
Show a query to retrieve the name and address of all employees who work forthe `Research` department.

SELECT **FNAME, LNAME, ADDRESS**
FROM **EMPLOYEDD, DEPARTMENT**
WHERE **DNAME** = `Research` AND DNUMBER = DNO. and results of this query are.

NNAME	LNAME	ADDRESS
Jone	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

Example:
Show how to make a query on database of figure given above (Employee database) so that for every project located in 'Stafford' it can list the project number, the controlling department number, and the department manager's last name, address, and birth date.

SELECT **PNUMER, DNUM, LNAME, ADDRESS, BDATE**
FROM **PROJECT, DEPARTMENT, EMPLOYEE**
WHERE **DNUM** = DNUMBER **AND** MGRSSN = SSN **AND** PLOCATION = 'Stafford';

which gives the following list.

PNUMBER	DNUM	LNAME	ADDRESS	BDATE
10	4	Wallace	291, Berry, Belaire, TX	1941-06-20
30	4	Wallace	291, Berry, Belaire, TX	1941-06-20

Example:
Consider database shown in Fig. 1. For each employee, it is desired to retrieve the employee's first and last name and the first and last name of his or her immediate supervisor. Make a query and show the results.

Solution.

SELECT **E.FNAME, E.LNAME, S.FNAME, S.L.NAME**
FROM **EMPLOYEE AS E, EMPLOYEE AS S**
WHERE **E.SUPERSSN = S.SSN;**

Which gives the following results

333445555 Headquarters

E.FNAME	E.LNAME	S.FNAME	S.LNAME
Jone	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Alicia	Zelaya	Jennifer	Wallace
Jennifer	Wallace	James	Borg
Ramesh	Narayan	Franklin	Wong
Joyce	English	Franklin	Wong
Ahmad	Jabbar	Jennifer	Wallace

Example:
For the database shown in Fig. 1 make a

query which selects all EMPLOYEE SSNs and show the results also.

SELECT SSN
FROM EMPLOYEE;

And the corresponding results are

SSN
123456789
333445555
999887777
987654321
666884444
453453453
987987987
999665555

Example:

With respect to database of fig. what will be the result of the following query.

SELECT SSN, DNAME
FROM EMPLOYEE,
DEPARTMENT;

The result of this query is

SSN	DNAME
123456789	Research
333445555	Research
999887777	Research
987654321	Research
666884444	Research
453453453	Research
987987987	Research
888665555	Research
123456789	Administration
333445555	Administration
888665555	Administration
999887777	Administration
123456789	Headquarters
987654321	Administration
666884444	Administration
453453453	Administration
987987987	Administration
999887777	Headquarters
987654321	Headquarters
666884444	Headquarters
453453453	Headquarters
987987987	Headquarters
888665555	Headquarters

Example:

With respect to database of Figure given above (Employee Database) show the results the following query is made

**SELECT *
FROM EMPLOYEE
WHERE DNO = 5;**

The results of the query are as follows.

John	B	Smith	12345678	1965-09-	731 Fondren,	M	3000	333445	5
Franklin	T	Wong	33344555	1955-	638 Voss, Houston,	M	4000	888665	5
Ramesh	K	Naraya	66688444	1962-09-	975 FkB Oak,	M	38000	333445	S
Joyce	A	Engfeh	45345345	1972-07-	5631 FUce. Houston.	F	25000	333445	5

Example:

For the database of figure given above (Employee Database) show the results for the following query.

SELECT ALL SALARY
FROM EMPLOYEE;

The results are as follows

SALARY
30000
40000
25000
43000
38000
25000
25000
55000

Example:

Consider database of Figure given above (Employee Database) and give the results corresponding to following query.

SELECT DISTINCT SALARY
FROM EMPLOYEE;

The results are as follows

SALARY
30000
40000
25000
43000
38000
55000

Example:

With regard to data base of fig 1, make a query to retrieve the name of all employees who do not have supervisors.

The required query is shown below along with its results.

SELECT FNAME, LNAME

FROM EMPLOYEE
WHERE SUPERSSN IN NULL;
And the results are
FNAME LNAME
James Borg

RELATIONAL DATABASE TERMINOLOGY

Table. EMP

EMPNO	NAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO.
7839	KING	PRESIDENT		17-Nov-81	5000		10
7689	BLACK	MANAGER	7839	1-May-81	2850		30
7782	CLARK	MANAGER	7839	9-Jun-81	2450		10
7666	JONES	MANAGER	7839	2-Apr-81	2975		20
7654	MARTIN	SALESMAN	7698	28-Sep-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-Feb-81	1600	300	30
7844	TURNER	SALESMAN	7698	8-Seo-81	1500	0	30
7900	JAMES	CLERK	7698	3-Dec-81	950		30
7521	WARD	SALESMAN	7698	22-Feb-81	1250	500	30
7902	FORD	ANALYST	7566	3-Dec-81	*3000		20
7369	SMITH	CLERK	7902	17-Dec-80	800		20
7788	SCOTT	ANALYST	7566	9-Dec-82	3000		20
7876	ADAMS	CLERK	7788	12-Jan-83	1100		20
7934	MILLER	CLERK	7782	23-Jan-82	1800		10

4.21 TERMINOLOGY USED IN A RELATIONAL DATABASE

A relational database can contain one or many tables. A table is the basic storage structure of an RDBMS. A table holds all the data necessary about something in the real world-for example, employees, invoices, or customers.

The slide shows the contents of the EMP table or relation. The numbers indicate the following:

1. A single row or tuple representing all data required for a particular employee. Each row in a table should be identified by a primary key, which allows no duplicate rows. The order of rows is insignificant; specify the row order when the data is retrieved.
2. A column or attribute containing the employee number, which is also the primary key. The employee number identifies a unique employee in the EMP table. A primary key must contain a value.
3. A column that is not a key value. A column represents one kind of data in a table; in the example, the job title of all the employees. Column order is insignificant when storing data; specify the column order when the data is retrieved.
4. A column containing the department number, which is also a foreign key. A foreign key is a column that defines how tables relate to each other. A foreign key refers to a primary key or a unique key in another table. In the example,
   DEPTNO uniquely identifies a department in the DEPT table.
5. A field can be found at the intersection of a row and a column. There can be only one value in it.
6. A field may have no value in it. This is called a null value. In the EMP table, only employees who have a role of salesman have a value in the COMM (commission) field.

4.22 TABLE SAL GRADE.

GRADE	LOSAL	HISAL	TABLE DEPT		
1	700	1200	DEPT NO	DNAME	LOC
2	1201	1400	10	ACCOUNTING	NEWYORK
3	1401	2000	20	RESERCH	DALLAS
4	2001	3000	30	SALES	CHICAGO
5	3001	9999	40	OPERATIONS	BOSTON

Fig. Structure and data for three tablet EM SAL GRADE

Example:
Output for the statements
SQL > SELECT *
2 FROM DPT;

DEPT NO	DNAME	LOC
10	ACCOUNTING	NEWYORK
20	RESERCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Example:
Output of the statements,

SQL >SELECT ename, hiredale, sal
2 >FROM emp;

ENAME	HIREDATE	SAL
KING	17-NOV-81	5000
BLACK	01-MAY-81	2850
CLARK	09-JUN-81	2450
JONES	02-APR-81	2975
MARTIN	28-SEP-81	1250
ALLEN	20-FEB-81	1600

14 rows selected

Example:
Show the output corresponding to following
SQL> SELECT ename, sal, salt 300, 12* (sal + 1000)

ENAME	SAL	SAL+300	12*(SAL+100)
KING	5000	5300	61200
BLACK	2850	3150	35400
CLARK	2450	2750	30600
JONES	2975	3275	36900
MARTIN	1250	1550	16200
ALLEN	1600	1900	20400

14 rows selected

Example:
Output for
SQL> SELECT ename, job, sal, comm.
FROM emp;

ENAME	JOB	SAL	COMM
KING	PRESIDENT	5000	
BLAKE	MANNER	2850	
.....			
TURNER	SALESMAN	1500	0
.....			

.....
14 rows selected

Example:
Output for the
SQL> SELECT ename, 12* sal + comm.
2 FROM emp
3 WHERE ename = `KING`;
ENAME 12 * SAL + COMM
KING
This is because the comm column in the arithmetic expression is null, the result is null.

Example:
Output corresponding to
SQL> SELECT ename, "Name",
2 Sal* 12 "ANNUAL SALARY"
3 FROM emp;
NAME ANNUAL SALARY

Example:
Output for
SQL > SELECT ename, / / job AS "Employees"
2 FROM emp;
EMPLOYEES
KINGPRESIDENT
BLAKEMANAGER CLARK
MANAGER JONES MANAGER
MARTIN SALESMAN
ALLENSALESMAN
...
13 Rows selected

Example:
Show output for
SQL> SELECT ename, / /':'//'/'// "Month Salary" = / / Sal Monthly
2 FROM emp;
MONTHLY
KING: | Month Salary = 5000
BLAKE: | Month Salary = 2850
CLARK: | Month Salary = 2450
JONES | Month Salary = 2975
....
14 rows selected

Example:
O/p for
SQL > SELECT depto no
2 FROM emp;
DEPT NO.
10
30
10
20
14 rows selected

Example:
O/p for
SQL> SELECTDISTINCTdepot no
2 FROM emp;

DEPT NO.
10

30
10
There are only three unique department numbers in the table.

Example:
O/p for
SQL> SELECT DISTINCT depot no, Job
2 FROM emp;

DEPT NO.	**JOB**
10	CLERK
10	MANAGER
10	PRESIDENT
20	ANALYST

……..
9 rows selected

5 FILE STRUCTURE

5.1 RECORDS AND RECORD TYPES

Data is usually stored in the form of **records.** Each record consists of a collection of related data **values** of **items,** where each value is formed of one or more bytes and corresponds to a particular field of the record. Records usually describe entities and their attributes. For example, an EMPLOYEE record represents an employee entity, and each field value in the record specifies some attribute of that employee, such as NAME,BIRTHDATE, SALARY, or SUPERVISOR. A collection of field names and their corresponding data types constitutes a **record type** or **record format** definition.

A data type, associated with each field, specifies the type of values a field can take.

The data type of a field is usually one of the standard data types used in programming.

These include numeric (integer, long integer, or floating point), String of characters (fixed-length or varying), Boolean (having 0 and 1 or TRUE and FALSE values only), and sometimes specially coded **date** and **time** data types,

5.1.1 The number of bytes required for each data type is fixed for a given computer System.

Integer may require 4 bytes, a long integer 8 bytes, a real number 4 bytes, a Boolean 1 byte, a data 10 bytes (assuming a format of YYYY-MM-DD), and a fixed-length string of k characters k bytes. Variable-length strings may require as many bytes as there are characters in each field value.

For example, an EMPLOYEE record type may be defined-using the C programming language notation-as the following structure:

```
struct employee {
        char name [3θ];
        char ssn [9];
        int salary;
        int job code;
    char department [2,0]
        };
```

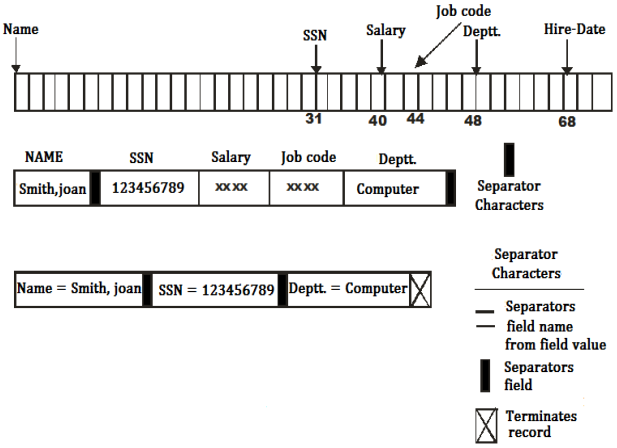
In recent database applications, it is required for storing data items that consist of large unstructured objects, which represent images, digitized video or audio streams, or free text.

These are referred to as BLOBs (Binary Large Objects). A BLOB data item is typically stored separately from its record in a pool of disk blocks, and a pointer to the BLOB included in the record.

5.1.2 FILES, FIXED –LENGTH RECORDS, AND VARIABLE-LENGTH RECORDS

A file is a sequence of records. In many cases, all records in a file are of the same record type. If every record in the file has exactly the same size (in bytes), the file is said to be made up to of **fixed-length** records. If different records in the file have different sizes, the file is said to be made up of variable **Length records.**

The fixed-length EMPLOYEE records in Figure have a record size of 71 bites. Every record has the same fields, and field lengths are fixed, so the system can identify the starting byte position of each field relative to the starting position of the record



5.1.3 THREE FIELD RECORD STORAGE FORMATS

For variable-length fields; each record has a value for each field, but we do not know the exact length of some field values. To determine the bytes within a particular record that represents each field, we can use special **separator** characters (such as? Or % $) - which do not appear of the field value-terminate variable-length fields or we can store the length in bytes of the field in the record, preceding the field value.

5.1.4 RECORD BLOCKING AND SPANNED VERSUS UNSPANNED RECORDS

The records of a file must be allocated to disk blocks because a block is the unit of data transfer between disk and memory. When the block size is larger than the record size, each block will contain numerous records. Although some files may have unusually large records that cannot fit in one block. Suppose that the block size is B bytes. For a file of fixed-length records or size R bytes, with $B \geq R$, we can fitbfr = floor[B/R] records per block, where the floor(x) (floor function) rounds down the numbers x to an integer. The value bfr is called the **blocking factor** of the file. In general, R may not divide B exactly we leave some unused space in each block equal to B- (bfr * R) bytes.

To utilize this unused space, we can store part of a record on one block and the rest on another. A **pointer** at the end of the first block points to the block containing the remainder of the record in case it is not the next consecutive block on disk. This organization is called **spanned,** because records can span more than one block. Whenever a record is larger than a block, we must use a spanned organization. If records are not allowed to cross block boundaries, the organization is called **un-spanned.**

5.1.4.1 ALLOCATING FILE BLOCKS ON DISK

There are several standard techniques for allocating the block of a file on disk. In contiguous allocation the file blocks are allocated to consecutive disk blocks. This makes reading the whole file very fast using double buffering, but it makes expanding the file difficult. In **linked-allocation** each file block contains a pointer to the next file block. This makes it easy to expand the file but makes it slow to read the whole file. A combination of the two allocates clusters of consecutive disk blocks, and the clusters are linked. Clusters are sometimes called **file segments or extents.** Another possibility to use **indexed allocation,** where one or more index blocks contain pointers to the actual file blocks. It is also common to use combinations of these techniques.

5.1.4.2 ORGANIZATION AND ACCESS METHOD

The difference between the terms file organization and access method is: A file organization refers to the organization of the data of file into records, blocks, and access structures; this includes the way records and blocks are placed on the storage medium and interlinked. An access method, on the other hand, provides a group of operations-such as those listed earlier-that can be applied to a file. In general, it is possible to apply several access methods to a file organization. Some access methods, though, can be applied only to files organized in certain ways. For example, we cannot apply an indexed access method to a file without an index.

5.1.4.3 DIFFERENT FILE ORGANIZATION

Three primary file organizations are there unordered, ordered and hashed. Unordered files require a linear search to locate records, but record insertion is very simple.

5.1.5 Files of Unordered Records (Heap Files)

In this simplest and most basic type of organization, records are placed in the file in order in which they are inserted, so new records are inserted at the end of the file. Such an organization is called a **heap** or **pire file.**

This organization is often used with additional access paths, such as the secondary indexes. It is also used to collect and store data records for future use.

Inserting a new record is very efficient: the last disk block of the file is copied into a buffer; the new record is added; and the block is then rewritten back to disk. The address of the last file lock is kept in the file header. However, searching for a record using any search condition involves a linear search through the file block by block-an expensive procedure. If only one record satisfies the search condition, then, on the average, a program will read into memory and search half the file block before it finds the record. For a file of blocks, this requires searching (b/2) blocks, on average. If no records on several records satisfy the search condition, the program must read and search all **blocks** in the file.

To delete a record, a program must first find its block, copy the block into a buffer, then delete the record from the buffer, and finally **rewrite the block back** to the disk. This leaves unused space in the disk block. Deleting a large number of records in this way, results in wasted storage space. Another technique used for record deletion is to have an extra byte or bit, called a **deletion marker,** stored with each record. A record is deleted by setting the deletion marker to a certain value. A different value of the marker indicates a valid (not deleted) record. Search programs consider only valid records in a block when conducting their search. Both of these deletion techniques require periodic reorganization of the file to reclaim the unused space of deleted records. During reorganization, the file blocks are accessed consecutively, the records are packed by removing deleted records.

After such reorganization, the blocks are filled to capacity once more. Another possibility is to use the space of deleted records when inserting new records, although this requires extra bookkeeping to keep track of empty locations.

5.1.6 Files of Ordered Records (Sorted Files)

We can physically order the records of a file on disk based on the value of one of their fields called the ordering field. This leads to an ordered or sequential file. If the ordering field is also a key field of the file-a field guaranteed to have a unique value in each record-then the field is called the ordering key for the file.

5.1.7 Ordered records have advantages over unordered files

1) First, reading the records in order of the ordering key values become externally efficient because no sorting is required.
2) Finding the next record from the current one in order of the ordering key usually requires no additional block accesses, because the next record is in the same block as the current one (unless the current record is the last one in the block).
3) Using a search condition based on the value of an ordering key field results in faster access when the binary search technique is used, this constitutes an improvement over linear searches, although it is not often used for disk files.

A binary search for disk files can be done on the block rather than on the records. Suppose that the file has b blocks numbered 1, 2,……, b; the records are ordered by ascending value of their ordering key field; and we are searching for a record whose ordering field value is K.

Inserting and deleting records are expensive operations for an ordered file because the records must remain

physically ordered. To insert a record, we must find its correct position in the file, based on its ordering field value, and then make space in the file to insect the record in that position. For a large file can be very time-consuming because, on the average, half the file blocks must be read and rewritten after records are moved among them. For record deletion, the problem is less severe if deletion makers and periodic reorganization are used.

5.1.8 Hashing Techniques

Another type of primary file organization is based on hashing, which provides very fast access to records on certain search conditions. This organization is usually called a hash **file.** The search condition must be an equality condition on a single field, called the hash field of the file. In most cases, the hash field is also a key field of the file, in which case it is called the hash key. The idea behind hashing is to provide a function h, called a hash function or randomizing function, that is applied to the hash field value of a record and yields the **address** of the disk block in which the record is stored, a search for the within the block can be carried out in a main memory buffer. For most records, we need only a single-block access to retrieve that record.

i) Internal Hashing

For internal files, hashing is typically implemented as a hash table through the use of an array records. Suppose that the array index range is from 0 to M - 1 then we have M slots whose addresses correspond to the array indexes. We choose a hash function that transforms the hash field value into an integer between 0 and M - 1. One common hash function is the h(K) = K mod M function, which returns the remainder of an integer hash field value K offer division by M, this value is then used for record address.

A collision occurs when the hash field value of a record that is being inserted hashes to an address that already contains a different record. In this situation, we must insert the new record in some other position, since its hash address is occupied. The process of finding another position is called **collision resolution.**

ii)External Hashing for Disk Files

Hashing for disk files is called external hashing. To suit the characteristics of disk storage, the target address space is made of buckets, each of which holds multiple records. A bucket is either one disk block or a cluster of continuous blocks. The hashing function maps a key into a relative bucket number, rather than assign an absolute block address to the bucket. A table maintained in the file header converts the bucket number into the corresponding disk block address.

The collision problem is less severe with buckets, because as many records as will fit in a bucket can hash to the same bucket without causing problems. However, we must make provisions for the case, where a bucket is filled to capacity and a new record being inserted hashes to that bucket. We can use a variation of chaining in which & pointer is maintained in each bucket to a linked list of overflow records for the bucket. The pointers in the linked list should be record pointers, which include both a block address and a relative record position within the block.

Hashing provides the fastest possible access for retrieving an arbitrary record given the value of its hash field.

5.1.9 OTHER PRIMARY FILE ORGANIZATIONS

Files of Mixed Records

The file organizations so far assume that all records of a particular file are of the same record type. The records could be of EMPLOYEES, PROJECTS, STUDENTS, or DEPARTMENTS, but each file contains records of only one type.

5.2 B-TREES & OTHER DATA STRUCTURES

Other data structures can be used for primary file organizations. For example, if both the record size and the number of records in a file are small, some DMBS's offer the option of a B-tree data structure as the primary files organization. Any data structure that can be adapted to the characteristics of disk devices can be used as a primary file organization for record placement on disk.

Ordered files shorten the time required to read records in order of the ordering field. The time required to search for an arbitrary record, given the value of its ordering key field, is also reduced if a binary search is used. However, maintaining the records in order makes insertion very expensive; thus the technique of using an unordered overflow file to reduce the cost of record insertion was discussed. Overflow records are merged with the master file periodically during file reorganization.

Hashing provides very fast access to an arbitrary record of a file, given the value of its hash key. The most suitable method for external hashing is the bucket technique, with one or more contiguous blocks corresponding to each bucket. Collisions causing bucket over-flow are handled by chaining. Access on any non hash field is slow, and so is ordered access of the records on any field.

5.2.1 INDEXING TECHNIQUES

Indexes are required to represent the sequence of stored record occurrences within their stored file. The, various indexing techniques used are:

1. Non-dense indexing

The file being indexed is divided into groups, with several record occurrences in each group, such that's the following conditions hold.

i) For any two groups, all the stored record occurrences in one precede all those in the other (with respect to the sequencing being imposed on the file).

ii) Within any one group, the file sequence is represented by physical contiguity.

The index contains one entry per / group, giving the highest value of the indexed field occurring in the group and a pointer to the start of the group. The term "non-dense" refers to the fact that the index does not contain an entry for every stored record occurrence in the indexed file.

2. Multilevel Indexing

A multilevel (tree structure) index can contain any number of levels, each of which acts as a non- dense index to the level below. The top level, contain a single entry.

B-trees: B-Tree is a particular form of multilevel for tree structure index. B-tree actually denotes the index set. The nodes of a B-tree do not normally contain the same number of value entries. They normally do contain a certain amount of free space. In general, a B-tree of order n has at least n but not more than 2n value entries at any given node (and, if it has K value entries, then it also has K + n pointers).

Advantage: The B-tree insertion/deletion algorithm guarantees that the tree will remain balanced.

3. Indexing on Field Combinations

It is possible to construct an index on the basis of the values of two or more fields in combination. In general, an index on the combination of n fields F_1, F_2, ……F_n(in that order) will also serve. As an index on F_1, as an index on the combination F_1 F_2 (or F_2 F_1), as an index on the combination F_1 F_2 F_3 (in any order), and so on.

4. Selectivity in the Index

It is not necessary to provide access via the index to every record occurrence in the indexed file. In some situations, it may be useful to have index entries for selected7

values only of the indexed field. For example, in an employee file with 95% employees having status exempt and 5% non-exempt, it would be very useful to have an index pointing to all non-exempt employees.

5. Symbolic Pointer Representation:

SRA-valued pointers can be replaced by the corresponding primary key values. This value can then be used to locate the corresponding record. An index using symbolic keys will clearly not need to be updated just because the indexed file has been reorganized. However, access via that index will be slower than access via an index that uses direct pointers.

CITY FILE

CITY	S#
London	S5
Paris	S1
	S4
Athens	S2
	S3

SUPPLIER FILE

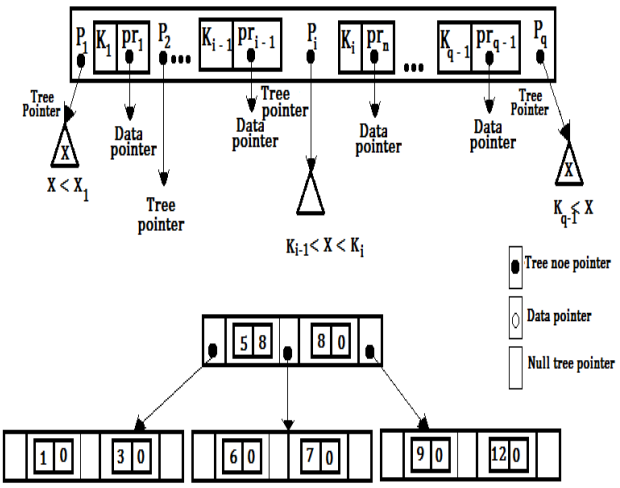
S#	SNAME	STATUS
S1	Smith	20
S2	Jones	10
S3	Blake	30
S4	Clark	20
S5	Adams	30

Fig indexing on city (symbolic pointers)

5.2.2 B-TREES

A **B-tree of order p,** when used as an access structure on a key field to search for records in a data file, can be defined as follows:

1. Each internal node in the B-tree which is given in below figure is of the form $<P_1, <K_1, Pr_1>, P_2, < K_2, Pr_2>,...<K_{q-1}, Pr_{q-1}>, P_q>$
   where $q \leq p$. Each P_i is a **tree pointer**-a pointer to another node in the **B-tree**.
   Each Pr_i is a data pointer-a pointer to the record whose search key field value is equal to K. (or to the data file block containing that record).
2. Within each node, $K_1 < K_2 < ... < K_{q-1}$.
3. For all search key field values X in the subtree pointed at by P_i (the i^{th} subtree, see given Figure below, we have:
   $K_{i-1} < X < K_i$ for $1 < i < q$; $X < K_i$ for $i = 1$; and $K_{i-1} < X$ for $i = q$.
4. Each node has at most p tree pointers.
5. Each node, except the root and leaf nodes, has at least [(p/2)] tree pointers. The root node has at least two three pointers unless it is the only node in the tree.
6. A node with q tree pointers, $q \leq p$, has q - 1 search key field values (and hence has q -1 data pointers).
7. All leaf nodes are at the same level. Leaf nodes have the same structure as internal nodes except that all of their tree pointers P_i are null



5.2.3 Fig. B-tree structures.

Figure illustrates a B-tree of order p=3. All search values K in the B-tree are unique because we assumed that the tree is used as an access structure on a key field. If we use a B-tree on a non key field, we must change the definition of the file pointers Pr_i to point to a block- or cluster – that contain the pointers to the file records.

B-tree starts with a single root node (which is also a leaf node) at level 0(zero). Once the root node is full with p-1 search key values and we attempt to insert another entry in tree, the root node splits into two nodes at level 1. Only the middle value is kept in the root node, and the rest of the

values are split evenly between the other two nodes. When a no root node is full and a new entry is moved to the parent node is split into two nodes at the same level, and the middle entry is moved to the parent node along with two pointers to the new split nodes. If the parent node is full, it is also split. Splitting can propagate all the way to the root node, creating a new level if the root is split.

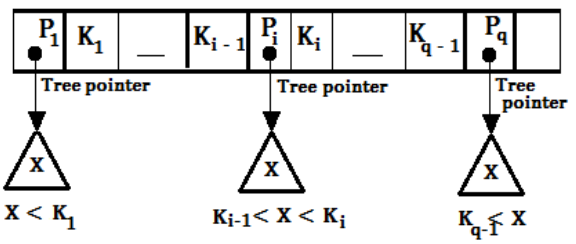
If deletion of a value causes a node to be less than half full, it is combined with its neighbouring nodes, and this can also propagate all the way to the root. Hence, deletion can reduce the number of tree levels. It has been shown by analysis and simulation that, after numerous random insertions and deletions on a B-tree, the nodes are approximately 69 percent full when the number of values in the tree stabilizes. This is also true of B*-trees. If this happens, node splitting and combining will occur only rarely, so insertion and deletion becomes quite efficient. If the number of values grows, the tree will expand without a problem-although splitting of nodes may occur, so some insertion will take more time.

B-trees are sometimes used as primary file organizations. In this case, whole records are stored within the B-tree nodes rather than just the <search key, record pointer>entries. This works well for files with a relatively small number of records, and a small record size. Otherwise, the fan-out and the number of levels become too great to permit efficient access. In summary, B-trees provide a multilevel access structure that is a balanced tree structure in which each node is at least half full. Each node in a B-tree of order p can have at most p - 1 search values.

5.2.4 B+-TREES

Most implementations of a dynamic multilevel index use a variation of the B-tree data structure called a B+-tree. In a B-tree, every value of the search field appears once at some level in the tree, along with a data pointer. In a B+-tree, data pointers are stored only at the leaf nodes of the tree; hence, the structure of leaf nodes differs from the structure of internal nodes. The leaf nodes have an entry for every value of the search field, along with a data pointer to the record (or to the block that contains this record) if the search field is a key field. For a non key search field, the pointer points to a block containing pointers to the data file records, creating an extra level of indirection. The leaf nodes of the B+-tree are usually linked together to provide ordered access on the search field to the record. These leaf nodes are similar to the first (base) level of an index. Internal nodes of the B+-tree correspond to the other levels of a multilevel index. Some search field values from the leaf nodes are repeated in the internal nodes of the B+-tree to guide the search. The structure of the internal nodes of a B'-tree of order p Figure is as follows:

1. Each internal node is of the form <P_1, K_1 P_2 K_2, ...P_{q-1}, ,K_{q-1}, P_q,> where $q \leq p$ and each P_i, is a tree pointer.
2. Within each internal node, K_1,< K_2,< ... <K_{q-1}.
3. For all search field values X in the subtree pointed at by P_i we have K_{i-1}, < $X \leq K_i$, for $1 < i < q$; $X \leq K_i$, for $i = 1$; and K_{i-1}, < X for $i = q$ (see Figure).
4. Each internal node has at most p tree pointers.
5. Each internal node, except the root, has at least [(p/2)] tree pointers. The root node has at least two tree pointers if it is an internal node.
6. An internal node with q pointers, $q \leq p$, has q - 1 search field values.



(a) Internal node of a B'-tree with q-1 search value

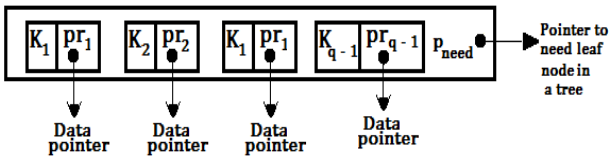


Figure : The nodes of a B+tree

The structure of the leaf nodes of a B+- tree of order p (Figure) is as follows:

1. Each leaf is of the from <<K_1, Pr_1>, <K_2, Pr_2>,…….,<K_{q-1}, Pr_{q-1}>, P_{next}>
   where $q \leq P$, each Pr_i each is a data pointer, and P_{next} points to the next leaf node of the B+-tree.
2. Within each leaf node, $K_1 < K_2 < ……. < K_{q-1}$, $q \leq p$.
3. Each Pr_i is a data pointer that points to the record whose search field value is K_i or to a file block containing the record (or to a block of record pointers that point to records whose search field value is K_i if the search field is not a key).
4. Each leaf node has at least [(p/2)] values.
5. All leaf nodes are at the same level.

The pointers in internal nodes are tree pointers to blocks that are tree nodes, whereas the pointers in leaf nodes are data pointer to the data file records or blocks-except for the P_{next} pointers which is a tree leaf nodes as a linked list using P_{next} pointers. This provides ordered access to the data records on the indexing field. A $P_{previous}$ pointer can also be included. For a B+-tree on a non key field, so the Pr pointers are block pointers to block that contain a set of record pointers to the actual records in the data file.

Because entries in the interval nodes of a B+-tree include search values and tree pointers without any data pointers, more entries can be packed into an internal node of a B+ -tree than for a similar B-tree. The can lead to fewer B+-tree levels, improving search time. Because the structures for internal and for leaf nodes of a B+-tree are different, the order P can be different. We will use p to denote the order for internal nodes and P_{leaf} to denote the order for leaf nodes which we define as being the maximum number of data pointers in a leaf node.

5.2.5 Variations of B-Trees and B+-Trees

In some cases, constraint 5 on the B-tree (or B+-tree), which requires node to be at least half full, can be changed to require each node to be at least two-thirds full. In this case the B-tree has been called a B*-tree. In general, some systems allow the user to choose a fill factor between 0.5 and 1.0, where the latter means that the B-tree (index) nodes are to be completely full. It is also possible to specify two fill factors for a B+-tree: one for the leaf level and one for the internal nodes of the tree. When the index is first constructed, each node is filled up to approximately the fill factors specified. Recently, investigators have suggested relaxing the requirement that a node be half full, and instead allow a node to become completely empty before merging, to simplify the deletion algorithm. Simulation studies show that this does not waste too much additional space under randomly distributed insertions and deletions.

5.3 DBMS

5.3.1 What is data model?

A collection of tools for describing data
Data relationships
Data semantics
Data constrains

5.3.2 System model tools:

Data flow diagram (DFD)
Hierarchical input process and output (HIPO)
State transition diagrams (STD)
Entity relationship (ER) diagrams

5.3.3 Entity-Relationship Model (ER Model)

A data model in which information stored in the database is viewed as sets of entities

and sets of relationships among entities and it is diagram-based representation of domain knowledge, properties etc….., but it is more intuitive and less mechanical

5.3.4 Components of ER model

a)Entity
b)Relationship
c) Attributes

5.3.5 Entity

Something that exists and can be distinguished from other entities.

Examples: Student entities with unique roll numbers Account entities with unique account Numbers

5.3.5.1 Entity set

A set of entities of the same type

- Example: All the student entities in a college
- Entity sets need not be disjoint. Example: a person entity could be in both the customer and employee sets.

5.3.5.2 Types of entities

Entities with physical existence

Example:
Student, customer, Book etc
Entities with Conceptual existence

Example:
Sale, University course etc

5.3.6 Relationship Set

A set of relationships of the same type

5.3.7 Attribute

A characteristic of an entity

- Example: A student entity might have attributes such as: Roll number, name, age, address etc.
- As all entities in an entity set have the same attributes, entity sets also have attributes –the attributes of the contained entities. The value of the attribute can be different for each entity in the set.

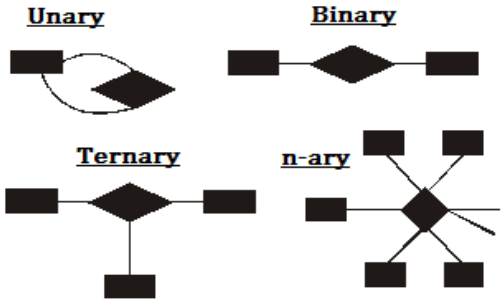
5.3.7.1 Domain of Attribute

A set of possible values for an attribute (thetype of attribute)

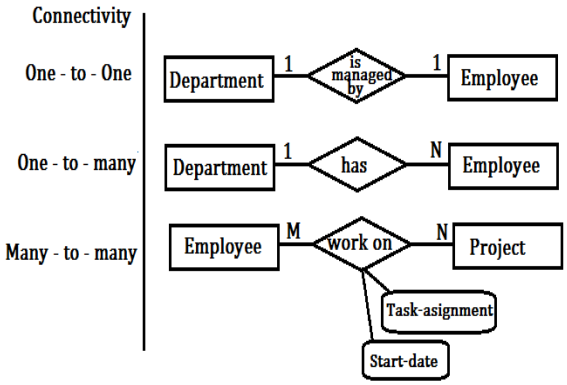
Example:
The domain of student name might be strings of some fixed length. The domain of roll number might be 10 digit positive integers or alphanumeric.

5.3.8 Relationship Degrees

The number of entity types associated with that relationship.



5.3.9 Types of relationships



5.3.10 Multiplicity

Multiplicity constrains the way that entities are related- It is a representation of the policies (or business rules) established by the user or enterprise. Multiplicity actually consists of two separate constraints

5.3.11 Cardinality

Which describes the maximum number of possible relationship occurrences for an entity participating in a given relationship type i.e. how many relationship instances is an entity permitted to be linked to.

5.3.12 Participation

Which determines whether all or only some entity occurrences participate in a relationship i.e. how is an entity linked to the relationship

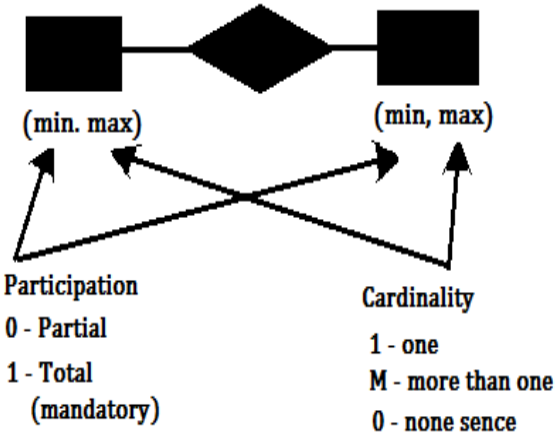
5.3.12.1 Total participation (indicated by double line):

Every entity set participates in at least one relationship in the relationship set

5.3.12.2 Partial participation

Some entities may not participate in any relationship in the relationship set

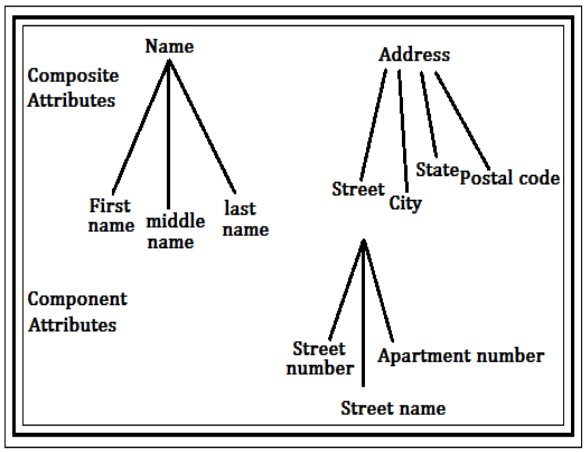
Note: Cardinality limits can also express participation constraints



5.3.12.3 Types or attributes

1. Simple and composite
2. Single-valued and multi-valued
3. Stored and derived attributes

Simple:
Position or salary attribute of faculty
Composite:
Address attribute composed of street, city and postcode attributes

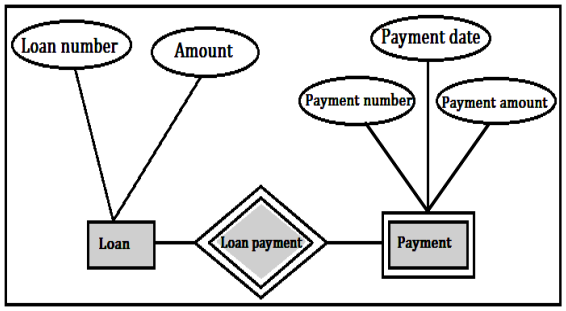


Single-valued:
Department No attribute of Department
Multi-valued:
Telephone No attribute of faculty
Stored:
Date of birth attribute of student
Derived:
Age attribute is derived from date of birth

5.3.13 Weak and a strong entity set

A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.
An entity set that does not have a primary key is referred to as a weak entity set. The existence of a weak relate to the identifying entity set via a total, one-to-many relationship set from the identifying to the weak entity set. Identifying relationship depicted using a double diamond.
The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set. The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak

entity set is existence dependent, plus the weak entity set's discriminator



We depict a weak entity set by double rectangles. We underline the discriminator of a weak entity set with a dashed line. Payment-number-discriminator of the payment entity set Primary key for payment – (loan-number, payment-number)

Note: The primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship. If loan-number were explicitly stored, payment could be made a strong entity, but then the relationship between payment and loan would be duplicated by an implicit relationship defined by the attribute loan-number common to payment and loan.

5.3.14 Reasons to have weak entities

We want to avoid the data duplication and consequent possible inconsistencies caused by duplicating the key of the strong entity.
Weak entities reflect the logical structure of an entity being dependent on another entity.

5.3.14.1 Weak:

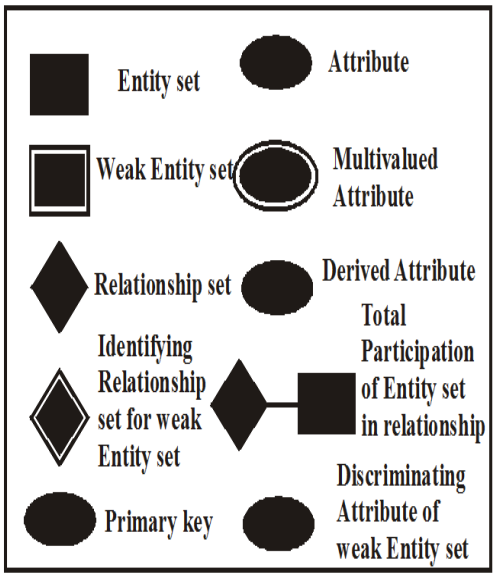
Entities can be deleted automatically when their strong entity is deleted. Weak entities can be stored physically with their strong entities

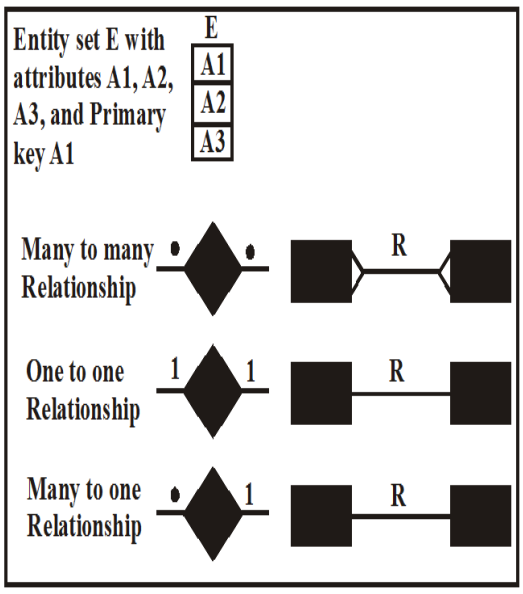
Examples
In a university, a course is a strong entity and a course-offering can be modelled as a weak entity. The discriminator of course-offering would be semester (including year) and section-number (if there is more than one section). If we model course-offering as a strong entity we would model course-number as an attribute. Then the relationship with course would be implicit in the course–number attribute

5.4 EXISTENCE DEPENDENCIES

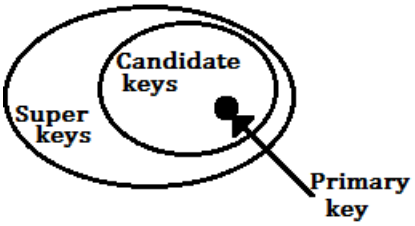
If the existence of entity x depends on the existence of entity y, then x is said to be existence dependent on y, y is a dominant entity, x is a subordinate entity. If y entity is deleted, then all its associated x entities must be deleted.

5.4.1 ER diagram symbols





5.4.2 Keys



Super key of an entity set is a set of one or more attributes whose values uniquely determine each entity and a key is a minimal super key: no subset of columns can determine entity.

A candidate key of an entity set is a minimal super key Student roll no is candidate key of student table, Book Accno is candidate key of Book table. Although several candidate keys may exist, one of the candidate keys is selected to be the primary key.

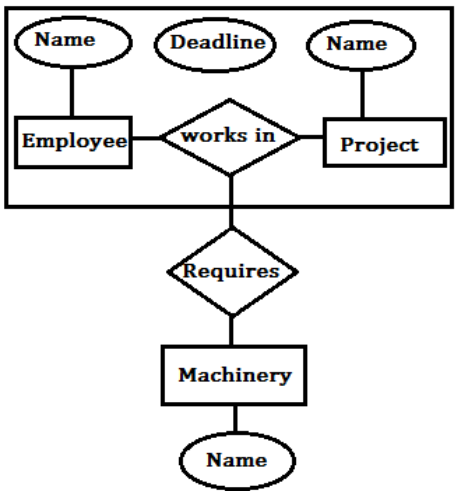
The combination of primary key of the participating entity sets forms a super key of a relationship set. (customer-id, account) is the super key of depositor.

NOTE: This means a pair of entity sets can have at most one relationship in a particular relationship set. E.g. if we wish to track all access-dates to each account by each customer, we cannot assume a relationship for each access. We can use a multi valued attribute though must consider the mapping cardinality of the relationship set when deciding, what are the candidate keys Need to consider semantics of relationship set in selecting the primary key in case of more than one candidate key.

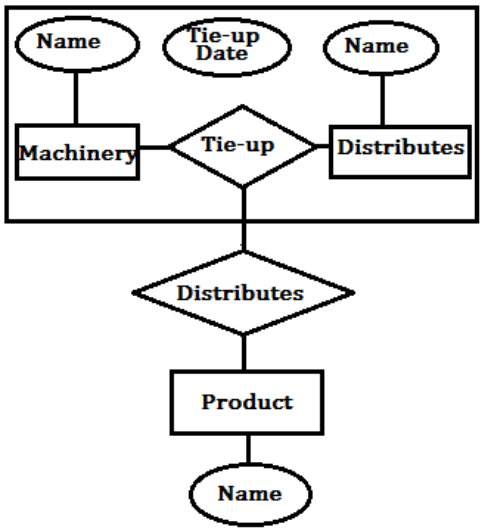
5.4.3 Aggregation

Aggregation is an abstraction through which relationships are treated as higher-level entities. Thus the relationship between entities A and B is treated as if it were an entity C. Some examples of this are:

a) Employees work for projects. An employee working for a particular project uses various machinery.



b) Manufactures have tie-ups with distributors to distribute products. Each tie-up has specified for it the set of products which are to be distributed.



5.4.4 Utility of E-R Model

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

5.4.5 E-R Design Decisions

The use of an attribute or entity set to represent object. Whether a real-world concept is best expressed by an entity set or a relationship set. The use of a ternary relationship versus a pair of binary relationships.
The use of a strong of weak entity set.
The use of specialization/ generalization – contributes to modularity in the design.
The use of aggregation – can treat the aggregate entity set as a single unit without concern for the details of its internal structure.

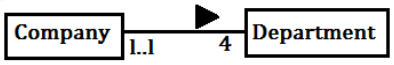
5.4.6 Tools to create E-R Diagrams

Several Computer Assisted Software Engineering (CASE) Tools exist to help create E-R Diagrams and the resulting physical database elements. Products include:
-IEW
-IEF
-DEFT
-ER-WIN
-Visio

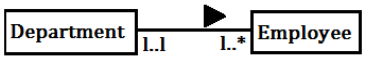
Exercises

1. Create an ER diagram for each of the following descriptions:

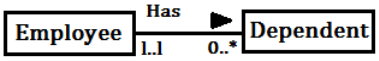
a) Each company operates four departments, & each department belongs to one company



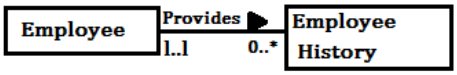
b) Each department in part (a) employs one or more employees, and each employee works for one department.



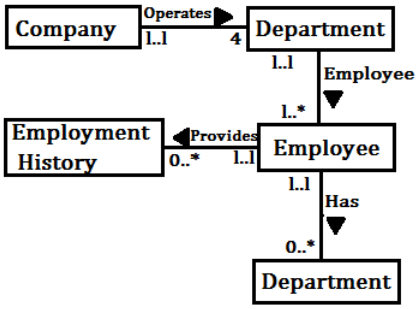
c) Each of the employees in part (b) may or may not have one or more dependants, & each dependant belongs to one employee.



d) Each employee in part (c) may or may not have an employment history.



e) Represent all the ER diagrams described in (a), (b), (c), and (d) as a single ER diagram.



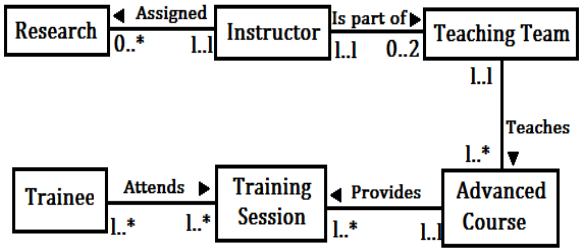
2. You are required to create a conceptual data model of the data requirements for a company that specializes in IT training. The Company as 30 instructors and can handle up to 100 trainees per training session. The Company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.

a) Identify the main entity types for the company

b) Identify the main relationship types and specify the multiplicity for each relationship. State any assumptions you make about the data.

c) Using your answers for (a) and (b), draw a single ER diagram to represent the data requirements for the company.

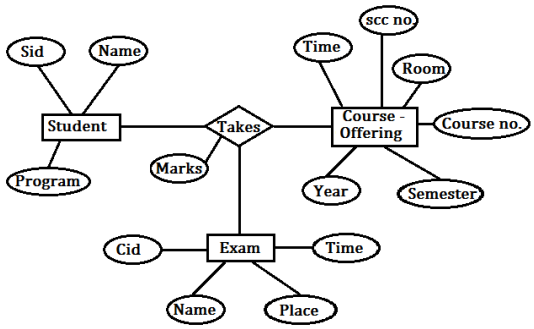
Answer:



3. Consider a database used to record the marks that students get in different exams of different course offerings.

Construct an E-R diagram that models exams as entities,

Answer:



4. A university registrar's office maintains data about the following entities: (a) courses including number, title, credits, syllabus, and prerequisites: (b) course offerings, including course number, year semester, section, number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modelled. Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints. Convert ER diagram into equivalent Relational tables

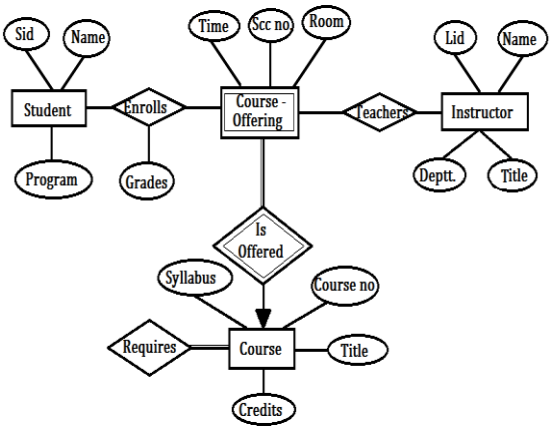
Answer:

In the answer given here, the main entity sets are student, course, course-offering, and instructor. The entity set course-offering is a weak entity set dependent on course. The assumptions made are:

a) A class meets only at one particular place and time. This E-R diagram cannot model a class meeting at different places at different times.

b) There is no guarantee that the database does not have two classes meeting at the same place and time.

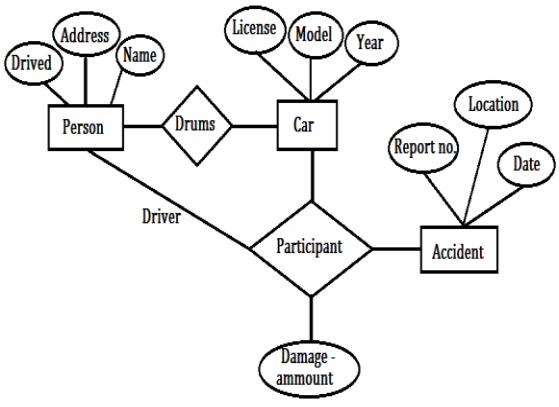
5.4.7 University registrar's tables:



Student (student-id, name, program)
Course (courseno, title, syllabus, credits)
Course-Offering (courseno,secho,year,semester,time, room)
Instructor (instructor-id, name, dept, title)
Enrolls(student-id,courseno,secno,semester,year, grade)
Teaches (courseno,secno,semester,year, instructor-id)
Requires (main course,prerequisite)

5. 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. Convert ER diagram into equivalent Relational tables.

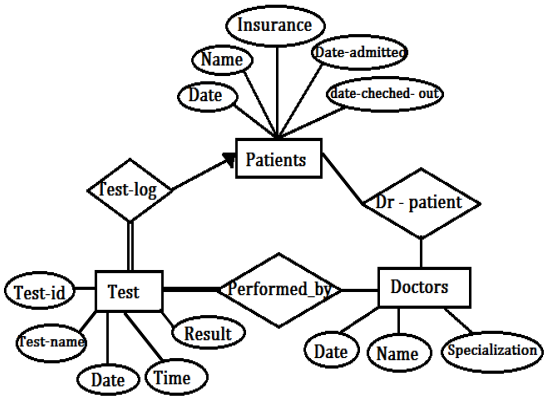
Answer :



Car insurance Tables
Person (driver-id, name, address)
Person (license, year, model)
Accident (report-number, data, location)
Participated (driver-id,license,report-number, damage-amount)

6. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Convert ER diagram into equivalent Relational tables.

Answer:



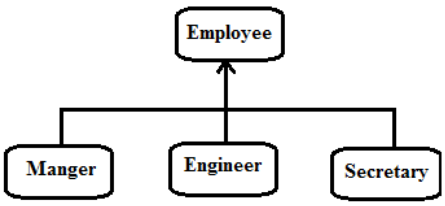
tables:
Patients (patient-id, name, insurance , date-admitted, date-check-out)
Doctors (doctor-id, name, specialization)
Test (testid, test name, date, time, and result)
Doctor-patient (patient-id,doctor-id)
Test-log (testid,patient-id)
Performed-by (testid,doctor-id)

GATE QUESTIONS

1 ER-MODEL

Q.1 It is desired to design an object-oriented employee record system for a company_ each employee has a name, unique Id and salary. Employees belong to different categories and their salary is determined by their category. The functions get Name, getId and compute salary are required. Given the class hierarchy below, possible locations for these functions are

1. getId is implemented in the super class
2. getId is implemented in the subclass
3. getId Name is an abstract function in the super class
4. getId Name is implemented in the super class
5. getId Name is implemented in the subclass
6. getId salary is an abstract function inthe super class
7. getId salary is implemented in the supercla.ss
8. getId salary is implemented in the subclass



Choose the best design.

a) 1, 4, 6, 7 b) 1, 4, 7
c)1, 3, 5, 6, 8 d) 2,5,8

[GATE – 2004]

Q.2 Consider the following entity relationship diagram (ERD), where two entities E1 and E2 have a relation R of cardinality 1: m

The attributes of E1 are A11, A12 and A13 where A11 is the key attribute. The attributes of E2 are A21, A22 and A23 is the key attribute and A23 is a multi-valied attribute. Relation R does not have any attribute. A relational database containing minimum number of table with each table satisfying the requirements of the third normal form (3NF) is designed from the above ERD. The number of table in the database is

a) 2 b) 3
c) 5 d) 4

[GATE-2004]

Q.3 Consider the entities 'hotel room', and 'person' with a many to many relationship 'lodging' as shown below:

If we wish to store information about the rent payment to be made by person (s) occupying different hotel rooms, then this information should appear as an attribute of

a) Person b) Hotel Room
c) Lodging d) None of these

[GATE – 2005]

Q.4 Let E_1 and E_2 be two entities in an E/R diagram with simple single valued attributes. R_1 and R_2 are two relationships between E_1 and E_2, where R_1 is one-to-many and R_2 is many-to-many. R_1, and R_2 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?

a) 2 b) 3
c) 4 d) 5

[GATE – 2005]

Q.5 The following table has two attributes A and C where A is the primary key

and C is the foreign key referencing A with on-delete cascade.

A	C
2	4
3	4
4	3
5	2
7	2
9	5
6	4

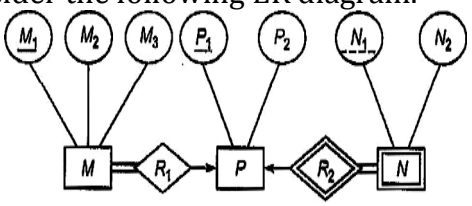
The set of all tuples that must be additionally deleted to preserve referential integrity when the tuple (2, 4) is deleted is:

a) (3, 4) and (6, 4)
b) (5, 2) and (7, 2)
c) (5, 2) (7, 2) and (9,5)
d) 1

[GATE – 2005]

Statements for Linked Answer Q.6 & Q.7

Consider the following ER diagram.



Q.6 The minimum number of tables needed to represent M, N, P, R_1,and R_2

a) 2 b) 3
c) 4 d) 5

[GATE – 2008]

Q.7 Which of the following is a correct attribute set for one of the tables for the correct answer to the above question?

a) $\{M_1, M_2, M_3, P_1\}$ b) $\{M_1, P_1, N_1, N_2\}$
c) $\{M_1, P_1, N_1\}$ d) $\{M_1, P_1\}$

[GATE – 2008]

Q.8 Given the basic ER and relational models, which of the following is incorrect?

a) An attribute of an entity can have more than one value
b) An attribute of an entity can be composite
c) In a row of a relational table anattribute can have more than one value
d) In a row of a relational table, an attribute can have exactly one value or a null value

[GATE – 2012]

Q.9 Consider an Entity-Relationship(ER) model in which entity sets E1and E2 are connected by an m : n relationship R12, E1 and E3 are connected by a 1 : n (1 on the side of E1 and n on the side of E3) relationship R13.

- E1 has two single-valued attributes a11 and a12 of which a11 is the key attribute.
- E2 has two single-valued attributes a21 and a22 of which a21 is the key attribute.
- E3 has two single-valued attributes a31 and a32 of which a31 is the key attribute.
- The relationships do not have any attributes.

If a relational model is derived from the above ER model, then the minimum number of relations that would be generated if all the relations are in 3NF is _____.

[GATE 2015]

Q.10 An ER model of a database consists of entity types A and B. These are connected by a relationship R which does not have its own attribute. Under which one of the following conditions, can the relational table for R be merged with that of A?

a) Relationship R is one – to- many and the participation of A in R is total
b) Relationship R is one- to- many and the participation of A in R is partial
c) Relationship R is many- to- one and the participation of A in R is total

d) relationship R is many – to – one and the participation of A in R is partial

[GATE 2017]

Q.11 In an Entity-Relationship (ER) model, suppose R is a many-to-one relationship from entity set E1 to entity set E2. Assume that E1 and E2 participate totally in R and that the cardinality of E1 is greater than the cardinality of E2.
Which one of the following is true about R?
a) Every entity in E1 is associated with exactly one entity in E2.
b) Some entity in E1 is associated with more than one entity in E2.
c) Every entity in E2 is associated with exactly one entity in E1.
d) Every entity in E2 is associated with at most one entity in E1.

[GATE-2018]

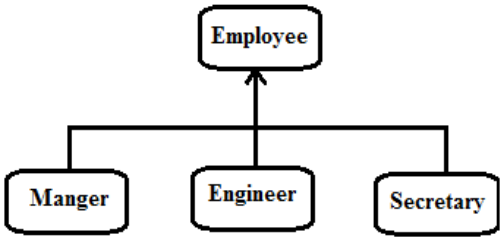
ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11
(a)	(b)	(c)	(b)	(c)	(b)	(a)	(c)	4	(c)	(a)

EXPLANATIONS

Q.1 **(a)**

ID is unique so; can be implemented in the super class.



Name is a function of super class.
Also, Name can be common/ duplicate, so must be implemented in super class. Salary is a function of super class. Also, salary is dependent on the category and hence is implemented in the subclass.

Q.2 **(b)**

One table for E1, two table for E2(A21, A22 and A23). We need to make a separate table for multi-valued attribute to satisfy minimum 1NF condition.
Now, Relation table can be merged with (A21, A23). Tables are:
a) E1(A11, A12, A13)
b) E21(A11,A21,A22) and
c) E22(A21, A23).
Number of tables = 3

Q.3 **(c)**

It is many to many. Rent cannot be an attribute of room or person entities lone. If depending on number of persons sharing a room the rent for each person for the room will be different.
Otherwise rent can be attribute of room.
Hence attribute is lodging.

Q.4 **(b)**

Given that
R1 is one to many and R2 is many to many.
The one to many relationships are represented with entity set from one side.
This normally happens as each entity in the entity set can be associated with at most one entity of the other.
Therefore, the table is not formed for R1 the tables are hence, formed for R2, E1 and E2.
So, there are total of 3 tables.

Q.5 **(c)**

If (2, 4) is deleted then 2 is the primary key but in (5, 2) and (7, 2), 2 is the foreign key so these must be deleted. The primary key for (5, 2) and (7, 2) is 5 and 7 respectively but in (9, 5), 5 is the foreign key so it is also deleted.

Q.6 **(b)**

Minimum No of tables required = 3 = {M, P, N} Since Cardinality of R1= N:1→ It will never become a table.
And Cardinality of R2 = 1:N→ It will never become a table.

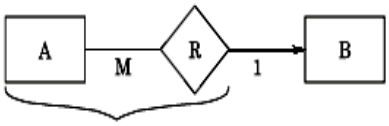
Q.7 **(a)**

Since P_2 depends on P_1, the only key value, so merging P and M in one result will lead to non-normalized tables. This needs to be removed without the loss to the information as P_1 and P_2 have primary key as P_1 and N_1 and N_2 has primary key as....N_1.
Therefore, the attribute set of table is $\{M_1, M_2, M_3, P_1\}$

Q.8 (c)
Option 'A' is correct as multivalued attribute e.g. phone no (attribute)
Option 'B' is also correct.
Option 'D' 'Null' values are allowed in a row of relational database (Null value are constraints only for primary key As in 1NF also we remove multivalued attribute
∴ Option 'C' is incorrect.

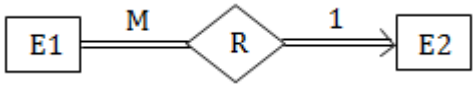
Q.9 (4)
Key in E1 is a11
Key in E2 is a21
Key in E3 is a31
Key in R12 will be a11+a21 since it is m:n relationship.
Since R13 is 1:n relationship, it cannot be a table. So the minimum number of tables needed is 4.

Q.10 (c)
Entity sets A,B
Relationship set R
Relation R merges with that of A.

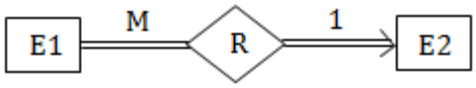


- Many to one relationship set can merge towards entity set 'A'.
- Partipation towards A side can be total partial.

Q.11 (a)



The M : 1 relationship holds between two entities E1 and E2, in which each tuple from E2 is in relation with many tuples of E1. One tuple from E1 is in relation with only one tuple of E2. It is given that participation from both the sides is total and the cardinality of E1 is greater than E2.



Therefore, every entity E1 is associated with exactly one entity in E2.

2 DATABASE DESIGN: FUNCTIONAL DEPENDENCIES & NORMALIZATION

Q.1 Given the following relation instance

X	Y	Z
1	4	2
1	5	3
1	6	3
3	2	2

Which of the following functional dependencies are satisfied by the instance?

a) XY→Z and Z→Y
b) YZ→X and Y→Z
c) YZ→X and X→Z
d) XZ→Y and Y→X

[GATE – 2000]

Q.2 Consider a schema R (A, B, C, D) with Functional dependencies A→B and C→DThen the decomposition of R into R_1 (AB) and R_2 (CD) is

a) dependency preserving and lossless join
b) lossless join but not dependency preserving
c) dependency preserving but not lossless join
d) not dependency preserving and not lossless joins

[GATE – 2001]

Q.3 R (A, B, C, D) is a relation. Which of the following does not have a lossless join, dependency preserving BCNF decomposition?

a) A→B, B→CD
b) A→B, B→C, C→D
c) AB → C, C → AD
d) A → BCD

[GATE – 2001]

Q.4 From the following instance of a relational schema R (A, B, C), we can conclude that

A	B	C
1	1	1
1	1	0
2	3	2
2	3	2

a) A functionally determines B and B functionally determines C
b) A functionally determines B and BS does not functionally determine C
c) B does not functionally determine C.
d) A does not functionally determine B & B does not functionally determine C

[GATE – 2002]

Q.5 Relation R is decomposed using a set of functional dependencies F, and relation S is decomposed using another set of functional dependencies G. One decomposition is definitely BCNF, the other is definitely 3NF, but it is not known which is to make a guaranteed identification, which one of the following tests should be used on the decompositions? (Assume that the closures of F and G are available)

a) Dependency-preservation
b) Lossless - join
c) BCNF definition
d) 3NF definition

[GATE – 2002]

Q.6 Relation R with an associated set of functional dependencies F is decomposed into BCNF. The redundancy (arising out of functional dependencies) in the resulting set of relations is

a) Zero
b) More than zero but less than that of an equivalent 3NIF decomposition
c) Proportional to the size of F
d) Indeterminate

[GATE – 2002]

Q.7 Consider the following functional dependencies in a database:

Data_of Birth→Age
Age→Eligibility
Name→Roll_number
Roll_number→Name
Course_number →Course_name
Course_number→Instructor
(Roll number; Course number) →Grade
The relation (Roll_number; Name, Date_of birth, Age) is

a) in second normal form but not in third normal form
b) in third normal form but not in BCNF
c) in BCNF
d) in none of the above

[GATE-2003]

Q.8 Consider the following relation schema pertaining to a student's database:
Student (rollno, name, address)
Enroll (rollno, courseno, coursename)
where the primary keys are shown underlined. The number of tuples in the student and Enroll tables are 120 and 8 respectively. What are the maximum and minimum number of tuples that can be present in (Student * Enroll), where '*' denotes natural join?

a) 8, 8 b) 120, 8
c) 960, 8 d) 960, 120

[GATE-2004]

Q.9 The relational schema Student Performance (name, course No, roll No, grade) has the following functional dependencies
Name, courser no → grade,
Roll No, course no → grade
Name → roll No
Roll No→ name
The highest normal form of this relation schema is

a) 2 NF b) 3 NF
c) BCNF d) 4 NF

[GATE – 2004]

Q.10 A relation Empdtl is defined with attributes empcode (unique), name, street, city, state and pincode. For any pincode, there is only one city and state. Also, for any given street, city and state, there is just one pincode. In normalization terms, Empdtl is a relation in

a) 1NF only
b) 2NF and hence also in 1NF
c) 3NF and hence also in 2NF and 1NF
d) BCNF and hence also in 3NF, 2NF an 1NF

[GATE-2004]

Q.11 A table has fields Fl, F2, F3, F4, F5 with the following functional dependencies
F1→F3, F2→F4, (F1, F2) →F5
In terms of Normalization, this table is in

a) 1 NF b) 2NF
c) 3 NF d) None of these

[GATE-2005]

Q.12 Which one of the-following statements about normal forms is false?

a) BCNF is stricter than 3NF
b) Lossless, dependency-preserving decomposition into 3NF is always possible
c) Loss less, dependency-preserving decomposition into BCNF is always possible
d) Any relation with two attributes is in BCNF

[GATE – 2005]

Q.13 Consider a relational schema R = (A, B, C, D, E, and H) on which the following functional dependencies hold: {A→B, BC→D, E→C, and D→ A}. What are the candidate keys of R?

a) AE, BE
b) AE, SE, DE
c) AEH, BEH, BCH
d) AEH, BEH, DEH

[GATE – 2005]

Q.14 In a schema with attributes A, B, C, D and E following set of functional dependencies are given
$A \rightarrow B$
$A \rightarrow C$
$CD \rightarrow E$
$B \rightarrow D$
$E \rightarrow A$
Which of the following functional dependencies is NOT implied by the above set
a) CD→AC b) BD→CD
c) BC→CD d) AC→BC
[GATE-2005]

Q.15 Consider the relations r_1(P,Q,R) and r_2 (R,S,T) with primary keys P and R respectively. The relation r_1 contains 2000 tuples and r_2 contains 2500 tuples. The maximum Size of the join $r_1 \bowtie r_2$ is :
a) 2000 b) 2500
c) 4500 d) 5000
[GATE-2006]

Q.16 Consider a relation R with five attributes V, W, X, Y, and Z.
The following functional dependencies hold: VY→W, WX→Z, and ZY→V.
Which of the following is a candidate key for R?
a) VXZ b) VXY
c) VWXY d) VWXYZ
[GATE-2006]

Q.17 The following functional dependencies are given AB→ CD, AF →D, DE →F, C →G, F→E, G →A
Which one of the following options is false?
a) $\{CF\}^+ = \{ACDEFG\}$
b) $\{BG\}^+ = \{ABCDG\}$
c) $\{AF\}^+ = \{ACDEFG\}$
d) $\{AB\}^+ = \{ABCDFG\}$
[GATE – 2006]

Q.18 Which one of the following statements is false?
a) Any relation with two attributes is in BCNF
b) A relation in which every key has only one attribute is in 2NF
c) A prime attribute can be transitively dependent on a key in a 3NF relation
d) A prime attribute can be transitively dependent on a key in a BCNF relation
[GATE – 2007]

Q.19 Consider the following implications relating to functional and multi valued dependencies given below, which may or may not be correct.
i) If A→→B and A→→C then A→BC
ii) If A→I3 and A→C then A→→4BC
iii) If A→→BC and A→B then A→C
iv) If A→BC and A→B then A→→C
Exactly how many of the above implications are valid?
a) 0 b) 1
c) 2 d) 3
[GATE-2007]

Q.20 Let R (A, B, C, D) be a relational schem a with the following functional dependencies:
A →B, B→C, C→D and D→B.
The decomposition of R into (A, B), (B, C), (B, D)
a) Gives a lossless join, and is dependency preserving
b) Gives a lossless join, but is not dependency preserving
c) Does not give a lossless join, but is dependency preserving
d) Does not give a lossless join and is not dependency preserving
[GATE-2008]

Q.21 Let R (A, B, C, D, E, P, G) be a relational schema in which the following functional dependencies are known to hold:
AB→CD,DE→P,C→E,P→C and B→G.
The relational schema R is
a) in BCNF

b) in 3NF, but not in BCNF
c) iri 2NF, but not in 3NF
d) not in 2NF

[GATE-2008]

Q.22 Consider the following relational schemas for a library database Book (Title, Author, Catalog no, Publisher, Year, Price)
Collection (Title, Author, Catalog no)
Within the following functional dependencies

1. Title Author →Catalog_no
2. Catalog no→Title Author Publisher Year
3. Publisher Title Year →Price

Assume {Author, Title} is the key for both schemas. Which of the following statements is true?

a) Both Book and Collection are in BCNF
b) Both Book and Collection are in 3NF only
c) Book is in 2NF and Collection is 3NF
d) Both Book and Collection are in 2NF only

[GATE – 2008]

Q.23 The following functional dependencies hold for relations R (A, B, C) and S (B, D, E)B → A, A → C
The relation R contains 200 tuples and the relation S contains 100 tuples. What is the maximum number of tuples possible in the natural join R ⋈ S

a) 100 b) 200
c) 300 d) 2000

[GATE – 2010]

Q.24 Consider a relational table with a single record for each registered student with the following attributes:

1. Registration_Number: Unique registration number for each registered student
2. UID: Unique Identity Number, unique at the national level for each citizen
3. BankAccount_Number: unique account number at the bank. A student can have multiple accounts or joint accounts. This attributes stores the primary account number
4. Name: Name of the Student
5. Hostel_room: Room number of the hostel

Which of the following options is incorrect?

a) BankAccount_Number is a candidate key
b) Registration_Number can be a primary key
c) UID is a candidate key if all students are from the same country
d) If S is a super key such that S UID ∩ is NULL then S UID is also a super key

[GATE – 2011]

Q.25 Which of the following is true?

a) Every relation in 3 NF is also in BCNF
b) A relation R is in 3NF if every non-prime attribute of R is fully functionally dependent on every key of R
c) Every relation in BCNF is also in 3 NF
d) No relation can be in both BCNF and 3NF

[GATE – 2012]

Common Data for Questions Q.26 and Q.27:
Relation R has eight attributes ABCDEFGH. Fields of R contain only atomic values. F = {CH→G, A→BC, B→CFH, E→A, F→EG} is a set of functional dependencies. So that F is exactly the set of FD's that hold for R.

Q.26 How many candidate keys do the relation R has?

a) 3 b) 4
c) 5 d) 6

[GATE – 2013]

Q.27 The relation R is
a) In INF, but not in 2NF
b) In 2NF, but not in 3NF
c) In 3NF, but not in BCNF
d) In BCNF

[GATE – 2013]

Q.28 The maximum number of super keys for the relation schema R(E, F, G, H) with E as the key is _____

[GATE- 2014]

Q.29 Given an instance of the STUDENTS relation as shown below:

Student ID	Student Name	Student Email	Student Age	CPI
2345	Shankar	shankar@math	X	9.4
1287	Swati	swati@ee	19	9.5
7853	Shankar	shankar@cse	19	9.4
9876	Swati	swati@mech	18	9.3
8765	Ganesh	ganesh@civil	19	8.7

For **(StudentName, StudentAge)** to be a key for this instance, the value X should NOT be equal to _____.

[GATE-2014]

Q.30 A prime attribute of a relation scheme R is an attribute that appears
a) In all candidate keys of R.
b) In some candidate key of R.
c) In a foreign key of R.
d) Only in the primary key of R.

[GATE -2014]

Q.31 Consider the relation scheme R=(E, F, G, H, I, J, K, L, M, N) and the given set of functional dependencies {EF→G, F→IJ, EH→KL, K→M, L→N} on R. What is the key for R?
a) {E,F} b) {E, F, H}
c) {E, F, H, K, L} d) {E}

[GATE- 2014]

Q.32 Given the following statements:
S1: A foreign key declaration can always be replaced by an equivalent check assertion in SQL.
S2: Given the table R(a,b,c) where a and b together form the primary key, the following is a valid table definition.

```
CREATE TABLES (
a INTEGER,
d INTEGER,
e INTEGER,
PRIMARY KEY (d),
FOREIGN KEY (a) references R)
```

Which one of the following statements is CORRECT?
a) S1 is TRUE and S2 is FALSE.
b) Both S1 and S2 are TRUE.
c) S1 is FALSE and S2 is TRUE.
d) Both S1 and S2 are FALSE.

[GATE-2014]

Q.33 Consider the following two statements:-
S1: Every table with two single-valued attributes is in 1NF, 2NF, 3NF, and BCNF
S2: AB→C, D→E, E→C is a minimal cover for the set of FD's AB→C, D→E, AB→E, E→C
Which one of the following is CORRECT?
a) S1 is TRUE and S2 is FALSE.
b) Both S1 and S2 are TRUE.
c) S1 is FALSE and S2 is TRUE.
d) Both S1 and S2 are FALSE.

[GATE-2014]

Q.34 Consider the relation X(P, Q, R, S, T, U) with the following set of functional dependencies
F = {{P, R}→ {S, T}, {P, S, U} →{Q, R}}
Which of the following is the trivial functional dependency in F^+, where F^+ is closure of F?
a) {P, R}→{S, T} b) {P, R}→{R, T}
b) {P, S}→{S} c) {P, S, U}→{Q}

[GATE- 2015]

Q.35 Which of the following is NOT a super key in a relational schema with attributes V, W, X, Y, Z and primary key V Y?
a) VXYZ b) VWXZ
c) VWXY d) VWXYZ

[GATE- 2016]

Q.36 A database of research articles in a journal uses the following schema.
(Volume, Number, StartPage, EndPage, Title, Year, Price)
The primary key is (Volume, Number, StartPage, EndPage) and the following functional dependencies exist in the schema.
(Volume, Number, StartPage, EndPage) → Title
(Volume, Number) → Year
(Volume, Number, Startpage, EndPage) → Price
The database is redesigned to use the following schemas.
(Volume, Number, StartPage, EndPage, Title, Price)
(Volume, Number, Year)
Which is the weakest normal form that the new database satisfies, but the old one does not?

a) 1NF b) 2NF
c) 3NF d) BCNF

[GATE -2016]

Q.37 The following functional dependencies hold true for the relational schema R{V,W,X, Y,Z}:

$V \rightarrow W$
$VW \rightarrow X$
$Y \rightarrow VX$
$Y \rightarrow Z$

Which of the following is irreducible equivalent for this set of functional dependencies?

a) $V \rightarrow W$, $V \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow Z$

b) $V \rightarrow W$, $W \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow Z$

c) $V \rightarrow W$, $V \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow X$, $Y \rightarrow Z$

d) $V \rightarrow W$, $W \rightarrow X$, $Y \rightarrow V$, $Y \rightarrow X$, $Y \rightarrow Z$

[GATE -2017]

Q.38 Consider the following four relational schemas. For each schema, all non-trivial functional dependencies are listed. The underlined attributes are the respective primary keys.

Schema I: Registration (<u>rollno</u>, courses)
Field 'courses' is a set-valued attribute containing the set of courses a student has registered for.
Non-trivial functional dependency:
Rollno → courses

Schema II: Registration (<u>rollno, courseid</u>, email)
Non-trivial functional dependencies:
rollno, coursed → email
email → rollno

Schema III: Registration (<u>rollno, courseid</u>, marks, grade)
Non-trivial functional dependencies:
rollno, coursed → marks, grade
marks → grade

Schema IV: Registration (<u>rollno, courseid</u>, credit)
Non-trivial functional dependencies:
rollno, coursed → credit
courseid → credit

Which one of the relational schemas above is in 3NF but not in BCNF?

a) Schema I b) Schema II
c) Schema III d) Schema IV

[GATE-2018]

ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(b)	(c)	(c)	(b)	(c)	(a)	(d)	(a)	(b)	(b)	(a)	(c)	(d)	(b)	(b)
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
(b)	(c)	(d)	(c)	(c)	(d)	(c)	(a)	(a)	(c)	(b)	(a)	8	19	(b)
31	**32**	**33**	**34**	**35**	**36**	**37**	**38**							
(b)	(b)	(a)	(c)	(b)	(b)	(a)	(b)							

EXPLANATIONS

Q.1 **(b)**

From the table given above, the B functional dependency is satisfied by the instance.

We can say this by the following:

Y is functionally dependent on Z.

→YZ is functionally dependent on X.

This is referred to as multi valued independency.

Q.2 **(c)**

We need to check two parameters for the solution

1. Lossless join
2. Dependency preserving

1. Lossless join

$R_1(AB) \wedge R_2(CD) \longrightarrow R_1(AB)$

also $R_1(AB) \wedge R_2(CD) \longrightarrow R_1(CD)$

But we know that Functional dependencies A→ B and C → D

Hence, the statement above is not possible.

$\longrightarrow R_1$ {ABCD) is not a lossless join.

2. Dependency preserving

Since,

$R_1(AB) \vee R_2(CD) \longrightarrow R(ABCD)$

It is dependency preserving.

Q.3 **(c)**

The solution exhibits as another case of multi valued dependency.

Here, the given relation is R (A,B,C, D).

This means all are dependent on each other.

Now, A→ Band B→ C

This implies, A→ C or we can say the AB→ C

Similarly,C →AD

Q.4 **(b)**

Let's consider A and B first.

A	B
1	1
1	1
2	3
2	3

Value of relation A is equal to the value of the relation B as A and B follows the same pattern. The value of A does not change where the value of B does not change.

Let's now consider B and C.

B	C
1	1
1	0
3	2
3	2

Value of relation B is not equal to the value of the relation B as B and C do not follow the same pattern. The value of C changes where the value of B does not change.

Q.5 **(c)**

Relation R is decomposed by functional dependency F.
Relation S is decomposed by functional dependency G.
One of the decompositions is in BCNF and other is in 3NF.
We cannot use 3NF definition since if a relation is in BCNF then it is also in 3NF.
We cannot use lossless-join because both decompositions are lossless.
We use BCNF definition so that 3NF fails on this test.

Q.6 (a)
According to the problem, R is associated with functional dependencies, so this clearly indicates that relation R is in 3NF, i.e., third normal form. The property of 3NF says that there are few or no error are left and therefore, redundancy (arising out of functional dependencies) in resulting set of relations is negligible or 0 in most cases.

Q.7 (d)
Relation (Roll_number, Name, Date-of-birth, Age)
Fd's which member of the relation
Date-of-birth→Age
Name→Roll_number
Roll_number→Name
Candidate keys {Name, date-of-birth}, {Rollnumber, date-of-birth}
Date-of-birth→Age is partial dependency which not allowed in 2NF. Highest NF of given relation 1 NF.

Q.8 (a)
Student (rno, name, address) {rno} unique.
Enroll (rno, cno, Cname) {rnocno} unique.
Each record of Enroll can be mapped one record of student based on natural join condition. So maximum 8 tuples in natural join and rno of Enroll foreign key references to student. So minimum also 8 tuples in natural join.

Q.9 (b)
Check for 1NF,
1NF says that the attributes should not be repeated. Here we can see that the attributes are not repeated. Condition is satisfied hence the relation schema is in 1 NF.
Check for 2NF,
2NF says that there should not be any partial dependency. Here it can be seen that there is no partial dependency. Condition is satisfied and hence the relation is in 2NF.
Check for 3NF,
Last two lines of the code show transitive dependency.
Name →Roll No
Roll No→ name
This is not against the conditions of 3NF and hence it is in 3NF.
Check for higher forms,
Since the conditions of 3NF are satisfied.

Q.10 (b)
It is in 2NF.
For 2NF, all non prime attributes should be fully functionally dependent on key. Key is empcode. Key contains only one attribute, hence no partial dependency, but there is transitive dependency. So it is not in 3 NF.

Q.11 (a)
key is F1F2
F1→F3,F2→F4 are partial dependencies.

Q.12 (c)
The statement false is Lossless; dependency-preserving
decomposition into BCNF is not always possible.

Lossless, dependency preserving decomposition is possible in certain cases only. It is important to note that BCNF is not dependency preserving in all cases but in some cases only.

Q.13 (d)

The keys AEH, BEH and DEH generate the (A,B,C,D,E,H) considering the functional dependencies
{ A → B, BC → D, E → C, and D → A}
So, AEH, BEH and DEH are the candidate keys of R because all of these generate {A, B, C, D, E, H}.

Q.14 (b)

Apply membership test for all the given Functional Dependencies.

1. CD → AC
   CD^+ = CDEAB
2. BD→CD
   BD^+ = BD

i.e. BD cannot derive CD and hence is not implied Similarly for rest two can be done.

Q.15 (b)

Relation r_1 contains 2000 tuples, and r_2 contains - 2500 tuples. Maximum common R values are possible for join of r_1 and r_2 is 2000.

Q.16 (b)

$(VY)^+$= VYW
$(VYX)^+$ = VYV,VXZ
We have a dependencies ZY→V & WX→Z
∴ VYX, ZYX and WXY are candidate keys.
⇒ VXY is candidate key for R.

Q.17 (c)

AF will give A and F. Also it gives D and E because AF→D and F→E. But we cannot get C and G.

Q.18 (d)

Statement (A) Any relation with two attributes is in BCNF is true as with the rules of BCNF, any relation with two attributes gets into Boyce code normal form.

Statement (B) A relation in which every key has only one attribute is 2NF is true as 2NF allows only the relation where a key has only one attribute.

Statement (C) A prime, attribute can be transitively dependent on a key in a 3NF relation is also true as,, prime attribute has to be transitively dependent on a key in 3NF.

Statement (D) A prime attribute can be transitively dependent on a key in a BCNF relation is false as this is a feature of 3NF and not BCNF.

Q.19 (c)

i) If A→→B and A→→C then A→BC not valid
ii) If A→B and A→C then A→BC, therefore A→→BC is also valid.
iii) If A→→BC and A→B, then A→C not valid
iv) If A→BC and A→B then A→C, therefore A→→C also valid.

Note: Every FD is a special case of multi-valued dependency.

Q.20 (c)

(A,B) and (B,C) have common attribute as B. Due to B→C, B is a key for (B,C).
Hence ABC can be loss lessly decomposed into (A,B) and (B,C).
(A, B, C) (B, D), common attribute is B.
B →D is a FD (via B→C, C→D), and hence B is a key for (B, D).
So, decomposition of (A, B, C, D) into (A, B, C) (B, D) is lossless. Thus the given decomposition is lossless.
The given decomposition is also dependency preserving:
A→B is is present in (A, B), B→C is present in (B, C), D→B is present in (B, D), and C→D is indirectly present via C→B in (B, C) and B→D in (B, D).

Q.21 (d)

Not in 2NF because here candidate key is AB and in FD's proper subset of C.K. determine the non prime attribute i.e. B→G.

Q.22 (c)

We will observe that if all the non-key fields are fully dependent on a whole key then a table would be formed in the Second normal form (2NF).

This table will be in the 3NF only when non-key fields are independent on a whole key than a table. Alternatively, Relation Book: transitivity dependency by 1, 2, 3 statements as given, hence the relation Book is in 2NF.

Relation Collection: it is in 3NF as there is fully transitive dependency.

Therefore Book is in 2NF and Collection is in 3NF.s

Q.23 (a)

B → A where R (A, B, C) and S (B, D, E), A → C

R contains 200 tuples and S contains 100 tuples.

Natural join ⋈=R ⋈ S = π[σRXS]

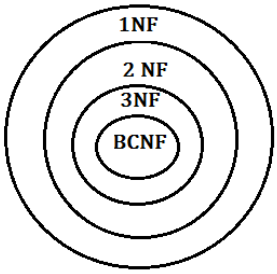
So, R = 200 tuples and S = 100 tuples

So, R ⋈ S = 100 [common is both or we can say distinct equality between all common attributes]

Q.24 (a)

Candidate key cannot be common. BankAccount_Number is not a candidate key. As per the question "A student can have multiple accounts or joint accounts. This attributes stores the primary account number". If two students have a joint account and if the joint account is their primary account, then BankAccount_Number value cannot uniquely identify a row.

Q.25 (c)



According to relation between the different normal forms i.e.,BCNF is more restricted than 3 NF and 3 NF is more restricted than 2 NF and so on.

Consider option (a):

The statement is given which is wrong because every relation which is in 3rd normal form it is not necessary that those will be in BCNF form. Hence, this is false.

Consider option (b):

When every non-prime attribute of R is fully functionally dependent on every key of R. Then, that relation is in 2 NF not in 3 NF.

Consider option (c):

Every relation which is in BCNF, then it is confirmed that it is also in 3 NF.

Consider option (d):

This statement is wrong because the relation can be in both BCNF and 3 NF.

Q.26 (b)

Relation contain 8 attributes ABCDEFGH

F={CH→G, A→BC, B→CFH, E→A, F→EG}

Candidate Key : AD, ED, BD, FD

∴Answer is (b) 4.

Q.27 (a)

Relations is in 1 NF, but not in 2NF because it is atomic and not in 2 NF because prime attributes of relation is {A, B, D, E, F]

A →BC⟹ written as (a) A →B

(b)A→c

In option (b) Violates definition of 2 NF, pot a prime attribute should

derive from, candidate key or not be derived from a prime attribute.
Option (c) Wrong answer - Because relation is not in 2NF.
Options (d) Relation is not in 2NF ⟹ it is not in 3NF & BCNF.
Answer is (a) in 1 NF, but not in 2NF.

Q.28 (8)
Maximum no. of possible super keys for a table with n attributes = 2^(n-1)

Q.29 (19)
(StudentName, StudentAge) is a key
Shankar X
Shankar 19
} here X should not be 19

Q.30 (b)
An attribute is called prime attribute if is part of any candidate key.

Q.31 (b)
The missing attributes on the right hand side of the FDs are E, F, H. These three attributes together can identify all other attributes. So {E, F, H} is the key.

Q.32 (b)
S1 is FALSE. A foreign key declaration cannot always be replaced by an equivalent check assertion in SQL.
S2: CREATE TABLE S (
a INTEGER,
d INTEGER,
e INTEGER,
PRIMARY KEY (d),
FOREIGN KEY (a) references R)
S**2** referenced attribute set (set of key attribute used for foreign reference) and referencing attribute set (set of foreign key attribute) must be same.

Q.33 (a)
S1: Table with two single valued attributes (A, B) will have key as either A or B or AB.
The FD's that are possible always will have key on the LHS. So it is in BCNF.
S2: The minimal cover cannot have AB→E. So S2 is false.

Q.34 (c)
And FD: X→Y is trivial if Y is subset of X.

Q.35 (b)
Super Key=Key+0 (or) more attributes.
Key = VY
Super Key = VY + more attributes
a) VXYZ = VY + XZ → Super Key
b) VWXZ = doesn't contain Key "VY" → Not a Super Key
c) VWXY = VY + WX → Super Key
d) VWXYZ = VY + WXZ → Super Key

Q.36 (b)
Old database → in 1NF
New Database→ in BCNF.
So the weakest normal form that the new database satisfies, but the old one does not is 2NF

Q.37 (a)
{V→W,VW→X,Y →V,Y→X,Y →Z}
We extraneous from VW→X
{V→W, V→X, Y→V, Y→X,Y→Z}
Y→X is redundant FD from above set
{V→W,V→X,Y→V,Y→Z} is minimal cover.

Q.38 (b)
Schema I:
Registration (rollno, courses) rollno → courses
For the given schema Registration 'rollno' is a primary key.
Left-side of the functional dependency is a superkey. so, Registration is in BCNF.

Schema II:

Registrstion (rollno, courseid, email)

rollno, courseid → email

email → rollno

From the given schema the candidate key is (rollno + courseid).

There is no part of the key in the left hand of the FD's so, it is in 2NF.

In the FD email→rollno, email is non-prime attribute but rollno is a prime attribute.

So, it is not a transitive dependency.

No transitive dependencies so, the schema is in 3NF.

But in the second FD email→rollno, email is not a superkey.

So, it is violating BCNF.

Hence, the schema Registration is in 3NF but not in BCNF.

Schema III:

Registration (rollno, courseid, marks, grade)

rollno, courseid → marks, grade

marks → grade

For the schema the candidate key is (rollno + courseid).

There are no part of the keys are determining non-prime attributes.

So, the schema is in 2NF.

In the FD marks → grade, both the attributes marks and grade are non-prime.

So, it is a transitive dependency.

The FD is violating 3NF.

The schema Registration is in 2NF but not in 3NF.

Schema IV:

Registration (rollno, courseid, credit)

rollno, courseid → credit

courseid → credit

The candidate key is (rollno + courseid).

In the FD, courseid → credit, courseid is part of the key (prime attribute) and credit is non-prime.

So, it is a partial dependency.

The schema is violating 2NF.

3 STRUCTURED QUERY LANGUAGE (SQL)

Q.1 In SQL, relations can contain null values, and comparisons with null values are treated as unknown. Suppose all comparisons with a null value are treated as false. Which of the following pairs is not equivalent?

a) X = 5 AND not (not (x = 5))
b) X = 5 AND x > 4 and x < 6, where x is an integer
c) X ≠5 AND not (x = 5)
d) None of the above

[GATE – 2000]

Q.2 Given relations r (w, x) and s(y, z), the result of Select distinct w, x From r, s Is guaranteed to be same as r, provided

a) r has no duplicates and s is non-empty
b) r and _s have no duplicates
c) s has no duplicates and r is non-empty
d) r and s have the same number of tuples

[GATE – 2000]

Q.3 Consider the following SQL query
select distinct $a_1, a_2, \ldots\ldots a_n$
from $r_1, r_2 \ldots\ldots r_m$
where ρ
For an arbitrary predicate ρ, this query is equivalent to which of the following relational algebra expressions?

a) $\prod_{a_1,a_2,\ldots,a_n} \sigma_\rho (r_1 \times r_2 \times \ldots\ldots \times r_m)$
b) $\prod_{a_1,a_2,\ldots,a_n} \sigma_\rho (r_1 \bowtie r_2 \bowtie \ldots\ldots \bowtie r_m)$
c) $\prod_{a_1,a_2,\ldots,a_n} \sigma_\rho (r_1 \cup r_2 \cup \ldots\ldots \cup r_m)$
d) $\prod_{a_1,a_2,\ldots,a_n} \sigma_\rho (r_1 \cap r_2 \cap \ldots\ldots \cap r_m)$

[GATE-2003]

Q.4 Consider the set of relations shown below and the SQL query that follows:

Students:(Roll_number,Name,Data_of_birth)
Courses:(Course_number,Course_name,Instructor)
Grades: (Roll_number, Course_number, Grade)
Select distinct Name
From Students, Courses, Grades
Where Students, Roll_number = Grades. Roll_number
And Courses.Instructor = Korth
And Courses.Course_number = Grades. Course_ number
And Grades. Grade = A
Which of the following sets is computed by the above query?

a) Names of students who have got an A grade in all courses taught By Korth
b) Names of students who have got an A grade in all courses
c) Names of students who have got an A grade in at least one of the courses taught by Korth
d) None of the above

[GATE – 2003]

Q.5 The employee information in a company is stared in the relation
Employee (<u>name,</u> sex, salary, dept, Name)
Consider the following SQL query:
Select dept Name
From Employee
Where sex = 'M'
Group by dept Name
Having avg (salary) >
(Select avg (salary) from Employee)
It returns the name of the department in which

a) The average salary is more than the average salary in the company
b) The average salary of male employees is more than the average salary of all male

employees in the company
c) Time average salary of male employees is more than the average salary of employees in the same department
d) The average salary of male employees is more than the average salary in the company

[GATE – 2004]

Q.6 A relational database contains two tables student and department in which student table has columns roll_no, name and dept_id and department table has columns dept_id and dept name. The following insert statements were executed successfully to populate the empty tables:

Insert into department values (1, Mathematics')
Insert into department values (2, 'Physics')
Insert into student values (1, 'Navin', 1)
Insert into student values (2,`Mukesh', 2)
Insert into student values (3, `Gita', 1)

How many rows and columns will be retrieved by the following SQL statement? Select * from student, department

a) 0 row and 4 columns
b) 3 rows and 4 columns
c) 3 rows and 5 columns
d) 6 rows and 5 columns

[GATE-2004]

Q.7 A table T1 in a relational database has the following rows & columns:

roll no.	marks
1	10
2	20
3	30
4	Null

The following sequence of SQL statements was successfully executed on table T1.

Update T1 set marks = marks +5
Select avg(marks) from T1

What is the output of the select statement?

a) 18.75 b) 20
c) 25 d) Null

[GATE-2004]

Q.8 Consider two tables in a relational database with columns and rows as follows:

Table : Student		
Roll_no	**Name**	**Dept_id**
1	ABC	1
2	DEF	1
3	GHI	2
4	JKL	3

Table : Department	
Dept_id	**Dept_name**
1	A
2	B
3	C

Roll_no is the primary key of the Student table. Dept_id is the primary key of the Department table and Student. Dept_id is a foreign key from Department. Dept_id

What will happen if we try to execute the following two SQL statements?

i) update Student set Dept_id = Null where Roll no = 1
ii) update Department set Dept_id = Null where Dept_id = 10

a) Both (i) and (ii) will fail
b) (i) will fail but (ii) will succeed
c) (i) will succeed but (ii) will fail
d) Both (i) and (ii) will succeed

[GATE-2004]

Q.9 A company maintains records of sales made by Its salespersons and pays them commission based on each individual's total sales made in a year. This data is maintained in a table with following schema:

salesinfo= (salespersonid, totalsales, commission)

In a certain year, due to better business results, the company

decides to further reward its sales persons by enhancing the commission paid to them as per the following formula:
If commission < = 50000, enhance it by 2%
If 50000 < commission < = 100000, enhance it by 4%
If commission > 100000, enhance it by 6%
The IT staff has written three different SQL scripts to calculate enhancement for each slab, each of these scripts is to run as a separate transaction as follows:
T_1 : Update salesinfo
Set commission = commission *1.02
Where commission < = 50000;
T2 : Update salesinfo
Set commission = commission * 1.04
Where commission> 50000 & commission
is < = 100000;
T3 : Update salesinfo
Set commission = commission * 1.06
Where commission > 100000;
Which of the following options of running these transactions will update the commission of all salespersons correctly?

a) Execute T1 followed by T2 followed by T3
b) Execute T2, followed by T3; T1 running concurrently throughout
c) Execute T3 followed by T2; T1 running concurrently throughout
d) Execute T3 followed by T2 followed by T1

[GATE-2005]

Q.10 In an inventory management system implemented at a trading corporation, there are several tables designed to hold all the information. Amongst these, the following two tables hold information on which items are supplied by which suppliers, and which warehouse keeps which items along with the stock-level of these items. Supply = (supplierid, itemcode)
Inventory = (itemcode, warehouse, stocklevel)
For a specific information required by the management, following SQL query has been written
Select distinct STMP.supplierid
From Supply as STMP
Where not unique (Select ITMP. supplierid
From Inventory, Supply as ITMP
Where STMP.supplierid = ITMP.supplierid
And ITMP.itemcode = Inventory.itemcode
And Inventory.warehouse = `Nagpur');
For the warehouse at Nagpur, this query will find all suppliers who

a) do not supply any item
b) supply exactly one item
c) supply one or more items
d) supply two or more items

[GATE-2005]

Q.11 The relation book (title, price) contains the titles and prices of different books. Assuming that no two backs have the same price, what does the following SQL query list?
Select title
From book as B
Where (select count (*)
From bock as T
Where T.pric>B.price) < 5

a) Titles of the four most expensive books
b) Title of the fifth most inexpensive book
c) Title of the fifth most expensive book
d) Titles of the five most expensive books

[GATE – 2005]

Directions for Question Q.12 to Q.13

Consider a database with three relation instances shown below. The

primary keys for the Drivers and cars relation are did and cid respectively and the records are stored in ascending order of these primary keys as given in the tables. No indexing is available in the database.

D: Drivers relation

did	dname	Rating	age
22	Karthikeyan	7	25
29	Salman	1	33
31	Boris	8	55
32	Amoldt	8	25
58	Schumacher	10	35
64	Sachin	7	35
71	Senna	10	16
74	Sachin	9	35
85	Rahul	3	25
95	Ralph	3	53

R : Reserves relation

did	cid	day
22	101	10/10/2006
22	102	10/10/2006
22	103	08/10/2006
22	104	07/10/2006
31	102	10/11/2006
31	103	06/11/2006
31	104	12/11/2006
64	101	05/09/2006
64	102	08/09/2006
74	103	08/09/2006

C: Cars relation

cid	cname	colour
101	Renault	blue
102	Renault	red
103	Ferrari	green
104	Jaguar	red

Q.12 What is the output of the following SQL query?

```
select D.dname
from Drivers D
where D.did in (
        select R.did
        from Cars C, Reserves R
        where R.cid=C.cid and C.colour
        = 'red'
        intersect
        select R.did
        from Cars C, Reserves R
        where
        R.cid=C.cid and C.colour
        = 'green')
```

a) Karthikeyan, Boris
b) Sachin, Salman
c) Karthikeyan, Boris, Sachin
d) Schumacher, Senna

[GATE-2006]

Q.13 Let n be the number of comparisons performed when the above SQL query is optimally executed. If linear search is used to locate a tuple in a relation using primary key, then n lies in the range

a) 36 - 40 b) 44 - 48
c) 60 - 64 d) 100 - 104

[GATE-2006]

Q.14 Consider the relation enrolled (student, course) in which (student, course) is the primary key, and the relation paid (student, amount) where student is -the primary key. Assume no null values and no foreign keys or integrity constraints. Given the following four queries:

$Query_1$: select student from enrolled where student in (select student from paid)

$Query_2$: select student from paid where student in (select student from enrolled)

$Query_3$: select E. student from enrolled E, paid P where E. student = P. student

$Query_4$: Select student from paid where exists (Select * from enrolled where enrolled. student = paid. student)

Which one of the following statements is correct?

a) All queries return identical row sets for any database
b) $Query_2$ and $Query_4$ return identical row sets for all

databases but there exist databases for Which $Query_1$, and $Query_2$ return different row sets

c) There exist databases for which $Query_3$ returns strictly fewer rows than $Query_2$

d) There exist databases for which $Query_4$ will encounter an integrity violation at runtime

[GATE – 2006]

Q.15 Consider the relation account (customer, balance) where customer is a primary key and there are no null values. We would like to rank customers according to decreasing balance. The customer with the largest balance gets rank I. Ties are not broken but ranks are skipped: If exactly two customers have the largest balance they each get rank 1 and rank 2 is not assigned.

Query 1	Select A. Customer, Count (B. Customer) From account A, account B Where A. balance <=B. balance Group by A. customer
Query 2	Select A. Customer, 1 + Count (B. Customer) From account A, account B Where A. balance < B. balance Group by A. customer

Consider these statements about $Query_1$, and $Query_2$

1 $Query_1$, will produce the same row set as $Query_2$ for some but not all databases.

2 Both $Query_1$, and $Query_2$ are correct implementation of the specification.

3 $Query_1$, is a correct implementation of the specification but $Query_2$ is not.

4 Neither $Query_1$, nor $Query_2$is a correct implementation of the specification.

5 Assigning rank with a pure relational query takes less time than scanning in decreasing balance order assigning ranks using ODBC.

Which two of the above statements are correct?

a) 2 and 5 b) 1 and 3

c) 1 and 4 d) 3 and 5

[GATE – 2006]

Q.16 Consider the table **employee** (empld, name, department, salary) and the two queries Q_1, Q_2 below. Assuming that department 5 has more than one employee, and we want to find the employees who get higher salary than anyone in the department 5, which one of the statements is true for any arbitrary employee table?

Q_1: Select e.empld
From employee e
Where not exists
(Select*from employee s where s.department="5"ands.salary>=e.salary)

Q_2: Select e.empld
From employee e
Where e.salary > any
(Select distinct salary From employee s Where s.department= "5")

a) Q_1, is the correct query

b) Q_2 is the correct query

c) Both Q_1 and Q_2 produce the same answer

d) Neither Q_1, nor Q_2 is the correct query

[GATE – 2007]

Directions for Question Q.17 to Q.18:

Consider the following relational schema:

Student(school-id, sch-roll-no, sname, saddress)

School(school-id, sch-name, sch-address, sch-phone)

Enrolment(school-id sch-roll-no, erollno, examname)

ExamResult(erollno, examname, marks)

Q.17 What does the following SQL query output?

```
SELECT sch-name, COUNT (*)
FROM School C, Enrolment E, ExamResult R
WHERE E.school-id C.school-id
      AND
      E.examname = R.examname
      AND
      E.Erollno = R.erollno
      AND
      R.marks = 100 AND S.school-id IN
      (SELECT school-id FROM student GROUP BY school-id HAVING COUNT (*) > 200)
      GROUP By school-id
```

a) for each school with more than 200 students appearing in exams, the name of the school and the number of 100s scored by its students
b) for each school with more than 200 students in it, the name of the school and the number of 100s scored by its students
c) for each school with more than 200 students in it, the name of the school and the number of its students scoring 100 in at least one exam
d) Nothing; the query has a syntax error

Borrower	Bank_Manager	Loan_Amount
Ramesh	Sunderajan	100000.00
Suresh	Ramgopal	5000.00
Mahesh	Sunderajan	7000.00

[GATE-2008]

Q.18 Consider the following tuple relational calculus query.

{t|∃E ∈ Enrolment t=E.school-id ∧| {x| x Enrolment∧x.school-id = t ∧ (∃B ExamResult B.erollno = x.erollno ∧ B.examname = x.examname ∧ B.marks > 35)} |÷ |{x | x Enrolment A x.school-id = I * 100 > 35}

If a student needs to score more than 35 marks to pass an exam, what does the query return?

a) The empty set
b) schools with more than 35% of its students enrolled in some exam or the other
c) schools with a pass percentage above 35% over all exams taken together
d) schools with a pass percentage above 35% over each exam

[GATE-2008]

Q.19 Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X =1, Y =1) is inserted in the table. Let M x and M $_Y$ denote the respective maximum values of X and Y among all records in the table at any point in time. Using M_X+1, 2*M_Y +1 respectively. It may be noted that each time after the insertion, values of M_X and M_Y change. What will be the output of the following SQL query after the steps mentioned above are carried out?

```
SELECT Y FROM T WHERE X = 7;
```

a) 127 b) 255
c) 190 d) 257

[GATE – 2011]

Q.20 Database table by name Loan_Records is given below

What is the output of the SQL query?

```
SELECT Count(*)
FROM (
(SELECT Borrower, Bank_Manager
FROM Loan_Records) AS S
NATURAL JOIN
(SELECT
Bank_Manager,Loan_Amount
 FROM Loan_Records) AS T);
```

a) 3 b) 9
c) 5 d) 6

[GATE – 2011]

Common Data for Questions Q.21 and Q.22

Consider the following relations A, B and C:

A.

ID	NAME	AGE
12	Arun	60
15	Shreya	24
99	Rohit	11

B.

ID	NAME	AGE
15	Shreya	24
25	Hari	40
98	Rohit	20

C.

ID	Phone	Area
10	2200	02
99	2100	01

Q.21 How many tuples does the result of the following relational algebra expression contain? Assume that the schema of A∪B is the same as that of A. $(A \cup B) \bowtie_{A.Id>40 \vee C.Id<15} C$

a) 7 b) 4
c) 5 d) 9

[GATE-2012]

Q.22 How many tuples does the result of the following SQL query contain?

```
SELECT A. Id
FROM A
WHERE A. Age > ALL (SELECT B. Age
FROM B
WHERE B. Name = 'Arun')
```

a) 4 b) 3
c) 0 d) 1

[GATE – 2012]

Q.23 Which of the following statements are true about an SQL query?

P. An SQL query can contain a HAVING Clause even If it does not have a GROUP BY clause

Q. An SQL query can contain a HAVING clause only if it has a GROUP BY clause

R. All attributes used in the GROUP BY clause must appear in the SELECT clause S. Not all attributes used in the GROUP BY clause need to appear in the SELECT clause

a) P and R b) P and S
c) Q and R d) Q and S

[GATE – 2012]

Q.24 Given the following schema:-

Employees (emp-id, first-name, last-name, hire-date, dept-id, salary) departments(dept-id, dept-name,manager-id, location-id)

You want to display the last names and hire dates of all latest hires in their respective departments in the location ID 1700. You issue the following query:-

```
SQL> SELECT last-name, hire-date
FROM employees
WHERE   (dept-id, hire-date)IN
        (SELECTdept-id,
        MAX(hire-date)
        FROM employees JOIN
        departments USING
        (dept-id)
        WHERElocation-id= 1700
        GROUP BY  dept-id);
```

What is the outcome?

a) It executes but does not give the correct result.

b) It executes and gives the correct result

c) It generates an error because of pair wise comparison

d) It generates an error because the GROUP BY clause cannot be used with table joins in a sub query.

[GATE-2014]

Q.25 SQL allows duplicate tuples in relations, and correspondingly defines the multiplicity of tuples in the result of joins. Which one of the following queries always gives the same answer as the nested query shown below:

Select * from R where a in (select S.a from S)

a) Select R.* from R, S where R.a=S.a
b) Select distinct R.* from R, S where R.a=S.a
c) Select R.* from R, (Select distinct a from S) as S1 where R.a=S1.a
d) Select R.* from R, S where R.a=S.a and is unique R

[GATE-2014]

Q.26 What is the optimized version of the relation algebra expression $\pi_{A1}(\pi_{A2}(\sigma_{F1}(\sigma_{F2}(r))))$, where Al, A2 are sets of attributes in r with A1⊂A2 and F1,12 are Boolean expressions based on the attributes in r ?

a) $\pi_{A1}(\sigma_{(F1 \wedge F2)}(r))$
c) $\pi_{A1}(\sigma_{(F1 \vee F2)}(r))$
b) $\pi_{A2}(\sigma_{(F1 \wedge F2)}(r))$
d) $\pi_{A2}(\sigma_{(F1 \vee F2)}(r))$

[GATE-2014]

Q.27 Consider the following relational schema:
employee(empId,empName,empDep)
customer(custId,custName,salesRep Id,rating)
salesRepId is a foreign key referring to empld of the employee relation. Assume that each employee makes a sale to at least one customer.
What does the following query return?

```
SELECT empName
FROM employee E
WHERE NOT EXISTS (SELECT custld
FROM customer C
WHERE C.salesRepId = E.empId
AND C.rating <>'GOOD');
```

a) Names of all the employees with at least one of their customers having a 'GOOD' rating.
b) Names of all the employees with at most one of their customers having a 'GOOD' rating.
c) Names of all the employees with none of their customers having a 'GOOD' rating.
d) Names of all the employees with all their customers having a 'GOOD' rating.

[GATE-2014]

Q.28 Consider the following relations:

Student	
Roll. No	Student Name
1	Raj
2	Rohit
3	Raj

Student		
Roll. No	**Course**	**Mark**
1	Maths	80
1	English	70
2	Maths	75
3	English	80
2	Physics	65
3	Maths	80

Consider the following SQL query.

```
SELECT S. Student_Name, sum (P.Marks)
FROM Student S, Performance P
WHERE S. Roll_No =P.Roll_No
GROUP BY S.Student_Name;
```

The number of rows that will be returned by the SQL query is _____.

[GATE 2015]

Q.29 SELECT operation in SQL is equivalent to

a) the selection operation in relational algebra
b) the selection operation in relational algebra, except that SELECT in SQL retains duplicates
c) the projection operation in relational algebra
d) the projection operation in relational algebra, except that SELECT in SQL retains duplicates

[GATE-2015]

Q.30 Consider the following relation **Cinema (theater, address, capacity)**

Which of the following options will be needed at the end of the SQL query

SELECT P1. address
FROM Cinema P1

Such that it always finds the addresses of theaters with maximum capacity?

a) WHERE P1. Capacity> = All (select P2. Capacity from Cinema P2)
b) WHERE P1. Capacity> = Any (select P2. Capacity from Cinema P2)
c) WHERE P1. Capacity> = All (select max(P2. Capacity) from Cinema P2)
d) WHERE P1. Capacity> = Any (select max (P2. Capacity) from Cinema P2)

[GATE 2015]

Q.31 Consider the following database table named water_schemes:

Water_schemes		
scheme_No	**district_name**	**capacity**
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	10
2	Churu	20
1	Dungargarh	10

The number of tuples returned by the following SQL query is _____.

With **total**(name, capacity) as
SELECT district_name, sum(capacity)
FROM water_schemes
GROUP BY district_name
With **total_avg** (capacity) as
SELECT avg(capacity)
FROM total
SELECT name
FROM **total**, **total_avg**
WHERE total.capacity >= total_avg.capacity;

[GATE-2016]

Q.32 Consider a database that has the relation schema EMP (Empld. Emp Name. and DeptName). An instance of the schema EMP and a SQL query on it are given below:

EMP		
EmpId	**Emp Name**	**Dept Name**
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

```
SELECT AVG(EC.Num)
FROM EC
WHERE(DepName,Num)IN
    (SELECTDepName, COUNT(EmpId)AS
                    EC(DepName,Num)
    FROM EMP
    GROUP BY DepName)
```

The output executing the SQL query is

[GATE- 2017]

Q.33 Consider the following tables T_1 & T_2

T_1	
P	**Q**
2	2
3	8
7	3
5	8
6	9
8	5
9	8

T_2	
P	**Q**
2	2
8	3
3	2
9	7
6	7
7	2

In table T_1 **P** is the primary key and **Q** is the foreign key referencing **R** in table T_2 with on-delete cascade and no update cascade. In table T_2,**R** is the primary key and S is the foreign key referencing **P** in table T_1 with on- delete set NULL and on- update cascade. In order to delete record (3,8) from table T_1, the number of additional records that need to be deleted from table T_1 is ____.

[GATE- 2017]

Q. 34 Consider the following database table named top-scorer.

top-scorer.		
Player	**Country**	**Gaols**
Klose	Germany	16
Ronaldo	Brazil	15
G Muller	Germany	14
Fontaine	France	13
Pele	Brazil	12
Klinsmann	Germany	11
Kocsis	Hungary	11
Batistuta	Argentina	10
Cubillas	Peru	10
Lato	Poland	10
Lineker	England	10
T Muller	Germany	10
Rahn	Germany	10

Consider the following SQL query:
SELECT to- player FROM top scorer as ta
WHERE ta. goals>ALL (SELECT tb. Goals
FROM top scorer as tb
WHERE tb. country ='Spain')
AND ta.goals>ANY(SELECT tc. Goals
FROM top scorer as tc
WHERE tc.country='Germany')
The number of tuples returned by the above SQL query is _____.

[GATE- 2017]

Q.35 Consider the following two tables and four queries in SQL.
Book (isbn, bname), Stock (isbn, copies)
Query 1: SELECT B.isbn, S.copies FROM Book B INNER JOIN Stock S ON B.isbn = S.isbn;

Query 2: SELECT B.isbn, S.copies FROM Book B LEFT OUTER JOIN Stock S ON B.isbn = S.isbn;

Query 3: SELECT B.isbn, S.copies FROM Book B RIGHT OUTER JOIN Stock S ON B.isbn = S.isbn;

Query 4: SELECT B.isbn, S.copies FROM Book B FULL OUTER JOIN Stock S ON B.isbn = S.isbn;

Which one of the queries above is certain to have an output that is a superset of the outputs of the other three queries?
a) Query 1 b) Query 2
c) Query 3 d) Query 4

[GATE-2018]

ANSWER KEY:

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
(c)	(a)	(a)	(c)	(d)	(d)	(c)	(c)	(d)	(d)	(d)	(a)	(c)	(b)	(c)
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
(b)	(b)	(c)	(a)	(a)	(a)	(b)	(c)	(b)	(c)	(a)	(d)	2	(d)	(a)
31	**32**	**33**	**34**	**35**										
2	2.6	0	7	(d)										

EXPLANATIONS

Q.1 **(c)**

c) {x≠5 and (x=5) are not equivalent.
If all are null values x≠5 gives false but not (x=5) gives true.

a) x=5 & not (not (x=5)are equivalent)

b) x=5 & (x>4 & x<6) are equivalent.

Q.2 **(a)**

Distinct keyword tells that it will give duplicate entries only once and if r has duplicate entries than it will not show them more than once which tells us that if answer is to be same as r then r should not have any duplicate entries. Options c and d are completely wrong but in option b it's also not necessary that s must also have no duplicate entries thus the answer is option a.

Q.3 **(a)**

The given SQL is
select distinct $a_1, a_2, \ldots\ldots a_n$
from $r_1, r_2 \ldots\ldots r_m$
where ρ
All possible combination of tuples from r_1, r_2, r_m
is denoted by $r_1 \times r_2 \times \ldots\ldots \times r_m$
If P is a predicate then to select the all, condition is denoted by
$\sigma_p(r_1 \times r_2 \times \ldots \times r_m)$
If we wants to select only some tuples in the relation then composite expression for above SQL is

$$\prod_{a_1, a_2, \ldots, a_n} \sigma_\rho (r_1 \times r_2 \times \ldots\ldots \times r_m)$$

Q.4 **(c)**

The relations are as given,
Students: (Roll_number, Name, Date_of_birth)
Courses: (Course number, Course_name, Instructor)
Grades:(Roll_number, Course_number, Grade)
Now, the distinct name is to be selected, where

1. Name of the student is selected on the basis of the grade.
2. Instructor of the course in Korth.
3. Courses selected on the basis of grade.
4. Grade should be A .

The query thus, computed is Name of the students who gets A grade in at least one of the courses taught by Korth.

Q.5 **(d)**

SQL query will compute as follows

1. Selects the name from the department
2. Selects the male employee
3. Computes that The average salary of male employees is more than the average salary in the company.

Q.6 **(d)**

There is no specific joining condition specified, so it will retrieve Cartesian product of the tables. Number of rows = Product of number of rows in each relation = 3 * 2 = 6 Number of columns = Sum of number of columns=3+2 = 5.

Q.7 **(c)**

Update on null gives null.
Average function ignores null values.
So. Average will produce
(15+25+35)/3 = 25.

Q.8 **(c)**

(i) If we update in STUDENT table dept id = NULL, it will not cause any problem to referenced table.

(ii) If we set in DEPARTMENT table Dept_id = NULL, it will produce inconsistency because in

STUDENT table we still have the tuples containing the Dept_id= 1.

Q.9 (d)

T3 followed by T2 followed by Ti will be correct execution sequence other cases some people will get two times increment.

Example: Suppose T1 followed by T2.

If initial commission is 49500 then he is belonging to < 50000

So 49500*1.02 = 50490.

He is eligible in second category then 50490*1.04=52509.6.

So he will get increment two times. but he is eligible for only one slab of commission.

Q.10 (d)

Nested query ensures that for only those suppliers it returns true which supplies more than 1 item in which case supplier id in inner query will be repeated for that supplier.

Q.11 (d)

The computation is as follows

1. Select the name of the title
2. Title is selected from the book
3. Initiates a count
4. Consider book as T
5. Compute the title of five most expensive books

Q.12 (a)

For color= "Red", did= {22, 22, 31, 64}

For color = "Green", did = {22, 31, 74}

Intersection of Red and Green will give ={22, 31} which is Karthikeyan and Boris.

Q.13 (c)

Sub query:

Select R.did

FROM Cars C, Reserves R

WHERE R. Cid=C.Cid and R.color=lted':

If query Executes optimally

(i) Select R.did

FROM (Select Cid

From Cars

Wherecolor = Red)C, Reserve R

Where R.Cid = C.Cid

Number of comparisions for above query:

- 4 comparision for Red car selection.
- 20 comparision for R.Cid = C.Cid.

(ii) Select R.did

FROM (Select Cid

From Cars

Where color = Green)C, Reserve R

Where R. Cid = C.Cid

Number of comparisions for above query:

- 4 comparision for Green car selection.
- 20 comparision for R.Cid = C.Cid.

(iii)Intersection comparision:

{22, 22, 31, 31, 64} result of (i) sub query. {22, 31, 74} result of (ii) sub query. 6 comparision for intersection by using linear search to search element.

(iv)In operation of outer query: 20 comparisions are required. Total comparision = 24+14+6+20 = 64

Q.14 (b)

PAID	
Student	**Amount**
1	2000
Primary Key= Student	

ENROLLED	
Student	**Course**
1	MCA
1	MBA
Primary Key = Student + Course	

Query 1:- SELECT student FROM enrolled WHERE student in (SELECT Student FROM Paid); **→returns 2 rows {1, 1}**

Query 2 :- SELECT student FROM Paid WHERE student in (SELECT Student

FROM enrolled);
→**returns 1 row {1}**

Query 3 :-
```
SELECT e.student
FROM enrolled e, paid p
WHERE e.student = p.student;
```
→**returns 2 rows {1, 1}**

Query 4 :-
```
SELECT student
FROM paid
WHERE EXISTS
(SELECT *
FROM enrolled
WHERE
enrolled.student =
paid.student);
```
→**returns 1row {1}**

Final Result = Query 2 and Query 4 return identical row sets for all databases, but there exit databases for which Query 1 & Query 2 return different row sets on the relations.

Q.15 (c)

Query$_1$ where A balance⇐B balance ←this line computes the A group of customers and the balance is less than dr equal to the group B.

Query$_2$ where A balance < B balance→ this line computes the A group of customers and the balance is less than the group B.

Therefore, the row produced by query$_1$ and query $_2$ will be same but not in every case as the results of query$_1$ and query$_2$ differ in case where query$_1$ has the situation⇐ but query$_2$ has only <.

None of the query provides the correct implementation of the specification.

Q.16 (b)

As per given,

Employee (emplidname, department, salary)

This suggests that an employees is to be denoted using the employee Id, name, department, salary

Query$_1$ It selects an employee but does not compute the result as after the 'where not exists', it does not have statement to produce the result.

Query$_2$ It selects an employee who gets higher salary than anyone in the department 5.

Therefore, the query $_2$ is correct.

Q.17 (b)

If Select clause consist aggregate and non aggregate columns.

All non aggregate columns in the Select clause must appear in Group By clause.

But in this query Group by clause consists school-id instead of school-name.

Q.18 (c)

Q.19 (a)

For X = 7⟹ V =127;

X	Y
1	1
2	3
3	7
4	15
5	31
6	63
7	127

Q.20 (a)

	S	T	
Borrowe	Bank Manager	Bank Manager	Loan Amount
Ramesh	Sunderajan	Sunderajan	10000.00
Suresh	Ramgopal	Ramgopal	5000.00
Mahesh	Sunderajan	Sunderajan	7000.00

S ⋈ T

Borrowe	Bank Manager	Bank Manager	Loan Amount
Ramesh	Sunderajan	Sunderajan	10000.00
Ramesh	Sunderajan	Sunderajan	7000.00
Ramesh	Sunderajan	Sunderajan	5000.00
Mahesh	Sunderajan	Sunderajan	10000.00
Mahesh	Sunderajan	Sunderajan	7000.00

Count*(S ⋈ T)=S

Q.21 (a)

$A \cup B$

ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11
25	Hari	40
98	Rohit	20

$(A \cup B) \bowtie_{A.Id > 40 \vee C.Id < 15}$

ID	Name	Age	ID	Phone	Area
12	Arun	60	10	2200	02
15	Shreya	24	10	2200	02
99	Rohit	11	10	2200	02
25	Hari	40	10	2200	02
98	Rohit	20	10	2200	02
99	Rohit	11	99	2100	01
98	Rohit	20	99	2100	01

Q.22 (b)

SELECT A. Id
FROM A
WHERE A. Age > All (SELECT B.Age FROM B WHERE B. Name = 'Arun')

In the table B 'Arun' is not there, so there query in WHERE clause returns null set. If a subquery returns zero rows, the condition evaluates as true. So, all 3 rows will be evaluated as true.

Q.23 (c)

As per SQL standard HAVING clause allowed only if GROUP BY clause exists.

SELECT[DISTINCT] Atributes
FROM relations
[WHERE condition]
[GROUP BY attributes
[HAVING conditio]]

All attributes used in the GROUP BY clause must appear in the SELECT clause.

Q.24 (b)

The inner query produces last max hire-date in every department located at location id 1700. The outer query simply picks all pairs of inner query. Therefore, the query produces correct result.

Q.25 (c)

Option A: does not give same result as that of given query in the case where there are two records matched in S for a record in R.

Option B: does not give same result as that of given query in the case where there are two same rows in relation R.

Option C: Gives same result as that of given query.

Option D: Not an SQL query.

Q.26 (a)

i) $\pi_{A1}(\pi_{A2}(X)) = \pi_{A1}(X)$, since $A1 \subset A2$

ii) $\sigma_{F1}(\sigma_{f2}(X)) = \sigma_{F1 \wedge F2}(X)$

$\therefore \pi_{A1}(\pi_{A2}(\sigma_{F1}(\sigma_{f2}(r)))) = \pi_{A1}(\sigma_{F1 \wedge F2}(r))$

Q.27 (d)

SELECT empName FROM employee E WHERE NOT EXISTS

```
(SELECT Custld
FROM Customer C
WHERE C.SalesRepld = E.empld
AND C.rating<>
```

where, the SQL query in the box represents all customers having other than 'good' while the complete query represents name of all employees with all their customers having a 'good rating'.

Q.28 (2)

The following are the two rows:

Student_Name	Marks
Raj	310
Rohit	140

Q.29 (d)

$\prod$ in relation algebra is similar to SELECT in SQL but the only difference is that '$\prod$' gives distinct rows by eliminating duplicates but SELECT does not remove duplicates

Q.30 (a)

When the ALL condition is followed by a list, the optimizer expands the

initial condition to all elements of the list and strings them together with AND operators. When the ANY condition is followed by a list, the optimizer expands the initial condition to all elements of the list and strings them together with OR operators.

Q.31 (2)

WITH Clause in SQL → Used to create temperory tables.

```
With total(name, capacity)as
   SELECT     district_name,sum(capacity)
   FROM       water_schemes
   GROUP BY  district_name
```

↓

Tempory Table "total"

total	
name	capacity
Ajmer	20
Bikaner	40
Churu	30
Dungargarh	10

```
With total_avg(capacity)as
   SELECT     avg(capacity)
   FROM       total
```

↓

Tempory Table "total"

total_avg
capacity
25

```
SELECT     name
FROM       total, total_avg
EHERE      total, capacity>=total_avg.capacity,
```

↓

Main query on temporary tables

Total X total_avg				
name	capacity	capacity	Total capacity>= tital_avgcapacity	
Ajmer	20	25	FALSE	
Bikaner	40	25	TRUE	Raw selected
Churu	30	25	TRUE	Raw selected
Dungarga rh	10	25	FALSE	

Final Output = 2 Rows

Name
Bikaner
churu

Q.32 (2.6)

EMP		
EmpId	**Emp Name**	**Dept Name**
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

```
SELECT AVG(EC.Num)
FROM EC
WHERE(DepName,Num)IN
     (SELECTDepName, COUNT(EmpId)AS
                     EC(DepName,Num)
     FROM EMP
     GROUP BY DepName)
```

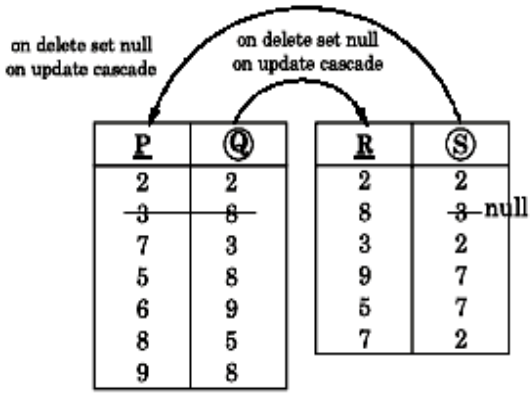
Result of inner query

EC

Dept Name	Number
AA	4
AB	3
AC	3
AD	2
AE	1

result of outer query : $\frac{13}{6} = 2.6$

Q.33 (0)



No other record need to delete because of deletion of (3,8) record from T_1

Q.34 (7)

Select ta. Player
FROM top scorer as ta
WHERE ta .Goals>ALL(SELECT tb. goals
FROM top scorer as tb
WHERE tb. country = 'Spain')
AND ta.goals > ANY (SELECT tc. goals
FROM top _scorer as tc
WHERE tc. Country = 'Germany'
Number of tuples in result 7.

Q.35 (d)

In SQL the **FULL OUTER JOIN** combines the results of both left and right outer joins and returns all (matched or unmatched) rows from the tables on both sides of the join clause. So, option (d) is correct.

4 RELATIONAL MODEL: RELATIONAL ALGEBRA & TUPLE CALCULUS

Q.1 Given the relations, Employee (name, salary, deptno), and Department (deptno, deptname, address)
Which of the following queries cannot be expressed using the basic relational algebra operations (σ , π, x, $\bowtie$,$\cap$,$\cup$, -)?
a) Department address of every employee
b) Employees whose name is the same as their department name
c) The sum of all employees' salaries
d) All employees of a given department

[GATE – 2000]

Q.2 Which of the following relational calculus expressions is not safe?
a) $\{t|\exists \cup \in R_1 (t[A] = u[A]) \wedge \neg \exists s \in R_2 (t[A] = s[A])\}$
b) $\{t|\forall u \in R_1 (u[A] = \text{“x”} \Rightarrow \exists s \in R_2 (t[A] = s[A] \wedge s[A] = u[A]))\}$
c) $\{t|\neg (t \in R_1)\}$
d) $\{t|\exists u \in R_1 (t[A] = u[A] = u[A] \wedge \exists s \in R_2 (t[A] = s[A])\}$

[GATE – 2001]

Q.3 With regard to the expressive power of the formal relational query languages, which of the following statements is true?
a) Relational algebra is more powerful than relational calculus
b) Relational algebra has the same power as relational calculus
c) Relational algebra has the same power as safe relational calculus
d) None of the above

[GATE – 2002]

Q.4 Consider the relation Student (name, sex, marks), where the primary key is shown underlined, pertaining to students in a class that has at least one boy and one girl. What does the following relational algebra expression produce?(Note la is the rename operator)
$\Pi_{name}(\sigma_{sex=female}(student)) - \Pi_{name}(student \bowtie_{(sex=female \wedge x=male \wedge marks \le n)} \rho_{n,xm}(student))$
a) names of girl students with the highest marks
b) names of girl students with more marks than some boy student
c) names of girl students with marks not less than some boy student
d) names of girl students with more marks than all the boy students

[GATE – 2004]

Q.5 Let R_1 (A , B , C) and R_2 (D , E) be two relational schemas, where the primary keys are shown underlined, and let C be a foreign key in R_1 referring to R_2. Suppose there is no violation of the above referential integrity constraint in the corresponding relational instances r_1 and r_2.Which one of the following relational algebra expressions would necessarily produce an empty relation?
a) $\prod_D (r_2) - \prod c (r_1)$
b) $\prod_c (r_1) - \prod_D (r_2)$
c) $\prod_D (r_2 \bowtie_{c \neq D} r_2)$
d) $\prod_c (r_1 \bowtie_{c = D} r_2)$

[GATE – 2004]

Q.6 Let r be a relational instance with schema R = (A, B, C, D). We define $r_1 = \prod_{A, B, C} (r)$ and $r_2 = \prod_{A, D} (r)$ and $r_2 = \prod_{A, D} (r)$ Let $s = r_1 * r_2$ where* denotes natural join. Given that the decomposition of r into r_1 and r_2 is

lossy, which one of the following is true?

a) $s \subset r$ b) $r \cup s = r$

c) $r \subset s$ d) $r * s = s$

[GATE – 2005]

Q.7 A table 'student' with schema (roll, name, hostel, arks), and another table 'hobby' with schema (roll, hobbyname) contains records as shown below:

Table Student

Roll	Name	Hostel	Marks
1798	Manoj Rathod	7	95
2154	Soumic Banerjee	5	68
2369	Gumma Reddy	7	86
2581	Pradeep Pendse	6	92
2643	Suhas Kulkarni	5	78
2711	Nitin Kadam	8	72
2872	Kiran Vora	5	92
2926	Manoj Kunkalikar	5	94
2959	Hemant Karkhanis	7	88
3125	Rajesh Doshi	5	82

Table Hobby

Roll	Hobbyname
1798	chess
1798	music
2154	musics
2369	Swimming
2581	cricket chess
2643	hockey
2643	volleyball
2711	football
2872	cricket
2926	photography
2959	music
3125	chess

The following SQL query is executed on the above tables:

Select hostel from student natural join hobby where marks > = 75 and roll between 2000 and 3000;

Relations S and H with the same schema as those of these two tables respectively contain the same information as tuples. A new relation S' is obtained by the following relational algebra operation:

$S' = \prod_{hostel} ((\sigma_{s.roll = H.roll}(\sigma_{marks > 75 \text{ and } roll > 2000 \text{ and } roll < 3000}(S)) \times (H))$

The difference between the number of rows output by the SQL statement and the number of tuples in S' is

a) 6 b) 4

c) 2 d) 0

[GATE-2005]

Q.8 Which of the following relational query languages have the same expressive power?

I. Relational algebra.

II. Tuple relational calculus restricted to safe expressions.

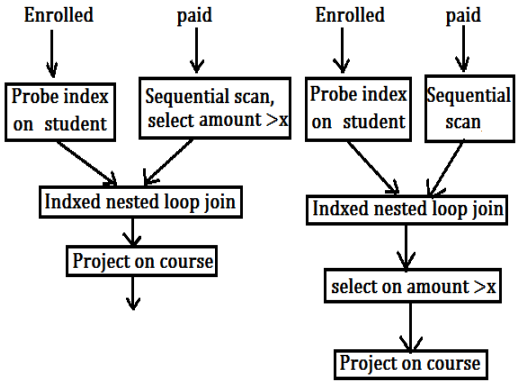
III. Domain relational calculus restricted to safe expressions.

a) II and III only b) I and II only

c) I and III only d) I, II and III

[GATE-2006]

Q.9 Consider the relation enrolled (student, course) in which (student, course) is the primary key, and the relation paid (student, amount) where student is the primary key. Assume no null values and no foreign keys or integrity constraints. Assume that amounts 6000, 7000, 8000, 9000 and 10000 were each paid by 20% of the students. Consider these query plans (Plan, on left, Plan2 on right) to "list all courses taken by students who have paid more than x"



A disk seek takes 4 ms, disk data transfer bandwidth is 300 MB/s and checking a tuple to see if amount is greater than x takes10 µs Which of the following statements is correct?

a) Plan1 and Plana will not output identical row sets for all databases
b) A course may be listed more than once in the output of Plan1, for some databases
c) For x=5000, Plan1, executes faster than Plan2 for all databases
d) For x=9000, Plan 1 execute slower than Plan 2 for all database

[GATE-2006]

Q.10 Consider a selection of the form $\sigma_{A \leq 100}(r)$, where r is a relation with 1000 tuples. Assume that the attribute values for A among the tuples are uniformly distributed in the interval [0, 500]. Which one of the following options is the best estimate of the number of tuples returned by the given selection query?

a) 50 b) 100
c) 150 d) 200

[GATE-2007]

Q.11 Consider the following relation schemas:
b-Schema = (b-name, b-city, assets)
a-Schema = (a-num, b-name, bal)
d-Schema = (c-name, a-number) Let branch, account and depositor be respectively instances of the above schemas. Assume that account and depositor relations are much bigger than the branch relation.
Consider the following query:
$\prod_{\text{c-name}}$ ($\sigma_{\text{b-city="Agra"} \wedge \text{bal<0}}$ (branch ⋈(account⋈depositor))
Which one of the following queries is the most efficient version of the above query?

a) $\prod_{\text{c-name}}(\sigma_{\text{bal} < 0}$ ($\sigma_{\text{b-city}}$ = "Agra" branch ⋈account) ⋈ depositor)
b) $\prod_{\text{c-name}}$ ($\sigma_{\text{b-city = "Agra"}}$ branch ⋈ ($\sigma_{\text{bal}< 0}$ account >a depositor))
c) $\prod_{\text{c-name}}$ ($\sigma_{\text{b-city = "Agra"}}$ branch ⋈ $\sigma_{\text{b - city= "Agra" bal < 0}}$ account ⋈depositor)
d) $\prod_{\text{c-name}}$ ($\sigma_{\text{b-city = "Agra"}}$ branch ⋈ ($\sigma_{\text{b-city= "Agra" bal < 0}}$ account ⋈depositor))

[GATE-2007]

Q.12 Consider the relation **employee** (name, sex, supervisor Name) with name as the key. Supervisor Name gives the name of the supervisor of the employee under consideration_ what does the following tunic relational calculus query produce **{e.name | employee(e) ∩ (∀x) [¬employee(x) U x. supervisorName ≠ e.name U x.sex="male"]}**

a) Names of employees with a male supervisor
b) Names of employees with no immediate male subordinates
c) Names of employees with no immediate female subordinates
d) Names of employees with a female supervisor

[GATE – 2007]

Q.13 Information about a collection of students is given by the relation stud info (studied, name, sex). The relation enrol (studied, coursed) gives which student has enrolled for (or taken) what course(s). Assume that every course is taken by at least one male and at least one female student. What does the following relational algebra expression represent?

$$\Pi_{\text{courseID}}\left(\Pi_{\text{studID}}\left(\sigma_{\text{sex='female'}}\left(\text{studInfo}\right)\right)\times \Pi_{\text{courseID}}\left(\text{enroll}\right)-\text{enroll}\right)$$

a) Courses in which al l the female students are enrolled
b) Courses in which a proper subset of
female students are enrolled
c) Courses in which only male students are enrolled
d) None of the above

[GATE – 2007]

Q.14 Let R and S be two relations with the following schemas.
R (P, Q, R_1, R_2, R_3)
S (P, Q, S_1, S_2)
Where {P, Q} is the key for both schemas. Which of the following queries are equivalent?
1. $\Pi_p(R \bowtie S$
2. $\Pi_p(R) \bowtie \Pi_p(S)$
3. $\Pi_p(\Pi_{P,Q}(R) \cap \Pi_{P,Q}(S))$
4. $\Pi_p(\Pi_{P,Q}(R) - (\Pi_{P,Q}(R) - \Pi_{P,Q}(S)))$

a) 1 and 2 b) 1 and 3
c) 1, 2 and 3 d) 1, 3 and 4

[GATE – 2008]

Q.15 Let R and S be relational schemas such that R={a, b, c} and S= {c}. Now consider the following queries on the database:

1. $\pi_{R-s}(r) - \pi_{R-s}(\pi_{R-s}(r) X S - \pi_{R-s,s}(r))$
2. $\{t \mid t \in \pi_{R-s}(r) \wedge \forall us(\exists v \in r(u = v[S] \wedge t = v[R-S]))\}$
3. $\{t \mid t \in \pi_{R-s}(r) \wedge \forall ur(\exists v \in s(u = v[S] \wedge t = v[R-S]))\}$
4. Select R.a, R.b
From R, S
Where R.c = S.c

Which of the above queries are equivalent?
a) 1 and 2 b) 1 and 3
c) 2 and 4 d) 3 and 4

[GATE – 2009]

Q.16 Consider the following relational query on the above database:
SELECT S.sname
FROM Suppliers S
WHERE S. sid NOT IN (SELECT C.sid
FROM Catalog C
WHERE C.pid NOT IN (SELECT P.pid
FROM Parts P
WHERE P.color <> 'blue))
Assume that relations corresponding to the above schema are not empty. Which one of the folly is the correct interpretation of the above query?
a) Find the names of all suppliers who have supplied a non-blue part
b) Find the names of all suppliers who have not supplied a non-blue part
c) Find the names of all suppliers who have supplied only blue parts
d) Find the names of all suppliers who have not supplied only blue parts

[GATE-2009]

Q.17 Assume that, in the suppliers relation above, each supplier and each street within a city has a unique name, and (sname, city) forms a candidate key. No other functional dependencies are implied other than those implied by primary and candidate keys. Which one of the following is TRUE about the above schema?
a) The schema is in BCNF
b) The schema is in 3NF but not in BCNF
c) The schema is in 2NF but not in 3 NF
d) The schema is not in 2NF

[GATE-2009]

Q.18 Suppose R_1, (A, B) and R_2 (C, D) are two relation schemas. Let r_1 and r_2 be the corresponding relation instances. B is a foreign key that refers to C in R_2. If data in r_1 and r_2 satisfy referential integrity constraints, which of the following is always true?
a) $\prod_B(r_1) - \prod_C(r_2) = \oslash$
b) $\prod_C(r_2) - \prod_B(r_1) = \oslash$
c) $\prod_B(r_1) = \prod_C(r_2)$
d) $\prod_B(r_1) - \prod_C(r_2) \neq \oslash$

[GATE – 2012]

Q.19 Consider the following relational schema.
Students(rollno:integer,sname:string)
Courses(courseno:integer,cname:string)
Registration(rollno: integer, courseno: integer, percent: real)
Which of the following queries are

equivalent to this query in English?
"Find the distinct names of all students who score more than 90% in the course numbered 107"

I) SELECT DISTINCT S.sname FROM Students as S, Registration as R WHERE R.rollno = S . roll no AND R course no=107 AND Rpercent>90

II) $\Pi_{sname}(\sigma_{courseno\ =107 \wedge percent>90})$ (Registration ⋈ students))

III) {T|∃S∈Students,∃R∈ Registration (S.rollno=R.rollno∧R.courseno=107∧R.percent>90∧T.sname=S.sname)}

IV) {<Sn>|$\exists S_R \exists R_P$(<S_R,S_n>∈ students∧<S_R,107,R_P>∈Registration ∧ R_P>90)}

a) I, II, III, and IV
b) I, II, and III only
c) I, II, and IV only
d) I, III, and IV only

[GATE – 2013]

Q.20 Consider a join (relation algebra) between relations r(R) and s(S) using the nested loop method. There are 3 buffers each of size equal to disk block size, out of which one buffer is reserved for intermediate results.
Assuming size(r(R)) < size(s(S)), the join will have fewer number of disk block accesses if

a) relation r(R) is in the outer loop.
b) relation s(S) is in the outer loop.
c) join selection factor between r(R) and s(S) is more than 0.5.
d) join selection factor between r(R) and s(S) is less than 0.5.

[GATE-2014]

Q.21 Consider the relational schema given below, where eId of the dependent is a foreign key referring to empId of the relation employee. Assume that every employee has at least one associated dependent in the dependent relation.

employee (empId, empName, empAge)
dependent(depId, eId, depName, depAge)
Consider the following relational algebra query:
$\prod_{empId}$(employee)-$\prod_{empId}$(employee ⋈$_{(empId=eId)\wedge(empAge\leq depAge)}$dependent)
The above query evaluates to the set of empIds of employees whose age is greater than that of

a) some dependent.
b) all dependents.
c) some of his/her dependents.
d) all of his/her dependents.

[GATE-2014]

Q.22 Consider two relations R1 (A, B) with the tuples (1, 5), (3, 7) and R2(A, C) = (1, 7), (4, 9).
Assume that R(A,B,C) is the full natural outer join of R_1 and R_2. Consider the following tuples of the form (A,B,C): a = (1, 5, null), b = (1, null, 7), c=(3, null, 9), d = (4, 7, null), e=(1, 5, 7), f=(3, 7, null), g=(4,null,9). Which one of the following statements is correct?

a) R contains a, b, e, f, g but not c, d
b) R contains all of a, b, c, d, e, f, g
c) R contains e, f, g but not a, b
d) R contains e but not f, g

[GATE-2015]

Q.23 Consider the relations *r(A,B)* and *s (B, C)*; where *s.B* is a primary key and *r.B* is a foreign key referencing *s.B*. Consider the query Q: r⋈($\sigma_{B<5}$(s))
Let *LOJ* denote the natural left outer-join operation. Assume that *r* and *s* contain no null values. Which one of the following is NOT equivalent to Q?

a) σB<5(r ⋈ s)
b) σB<5(r LOJ s)
c) r LOJ (σB<5(s))
d) σB<5(r) LOJ s

[GATE-2018]

ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(c)	(c)	(c)	(d)	(b)	(c)	(b)	(d)	(c)	(d)	(b)	(c)	(b)	(d)	(c)
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**							
(a)	(b)	(a)	(a)	(a)	(d)	(c)	(c)							

EXPLANATIONS

Q.1 **(c)**

The sum of all employees' salaries cannot be expressed using the given algebra operations since there is no command given for the sum of employees out of the algebra operations

Q.2 **(c)**

This option contradicts itself. We want a tuple t and t itself is in *j and we want other tuples than t by using negation so it's not safe.

Q.3 **(c)**

This is a rule of database theory that relational algebra has same power as safe relational calculus.

Q.4 **(d)**

Going with the given query, it means the following Name of the female student - name of the females who scored less than boys.

Therefore, it computes the name of the girl student with more marks than all the boy students.

Q.5 **(b)**

R_1 (A, B, C)

R_2 (D,E)

Given that, A is primary key of R_1 and D is primary key of R_2 and C is a foreign key in R_1, referring to primary key of R_2.

C→Child Column & D→ Parent Column

Result of Child – Parent is always Null.

So $\pi_C(r1)-\pi_D(r2)$ gives empty relation.

Q.6 **(c)**

R = (A, B, C, D)

$r_1 = \Pi_{A,B,C}(R)$ and $r_2 = \Pi_{A,D}(r)$

S = $r_1 * r_2$

The tuples in s are more than that of r. Also, s consists of all the tuples of r along with other tuples.

Therefore, the most appropriate relation is r ⊂ s.

Q.7 **(b)**

SQL query will return the following output.

Roll	Hostel	
2369	7	
2581	6	
2643	5	
2643	5	Duplicate Row is present in Hobby table
2872	5	
2926	5	
2959	7	

Total 7 rows are selected

In Relation Algebra, only distinct values of hostels are selected i.e. 5,6,7.

Therefore SQL row count RA row count = 7- 3 = 4.

Q.8 **(d)**

Given three relational query languages have same expressive power.

Q.9 (c)

Given that left hand side is $plan_1$ and right hand side is $plan_2$.

Plan1

1.Select records form paid

2.Joins them

Plan2

1.Joins records from paid

2.Records are checked

The seek time of disk is 4 ms and data transfer rate is 300 MB/s.

So, if x = 5000. Although the output remains the same but the $plan_1$ executes faster than $plan_2$ for all databases.

Q.10 (d)

Number of tuples in relation r = 1000

Each value of A is appeared twice, because uniformly distributed in [0, 500] $\sigma_{A\leq 100}(r)$ returns 200 tuples.

Q.11 (b)

As b is very small compared to a and d which is also taken in account.

$\sigma_{b\text{-}city="Agra"}$ branch which already filter city as agra making it small, and before that $\sigma_{bal<0}$ account $\bowtie$ depositor filter and give selected tables so $\bowtie$ between them will give same result and better one.

Q.12 (c)

We are given with the below relational calculus:

$(e.name \mid employee(e) \wedge \}$

$(\forall X)[\rightarrow employee(X) \vee X.supervisorName \neq X.sex="male"]$

$(e.name \mid employee(e) \wedge \}$ gives the name of the employee tells that employee does not have any female subordinate.

Q.13 (b)

$(\sigma_{sex="female"}(studinfo))$ tells that students are females.

$\Pi_{courseld}(Enroll))$ Gives the Id of the course in which to enroll.

$(\mathbf{\Pi}_{studid})$ tells regarding a proper subset of females that are enrolled.

Finally,

$\Pi_{courseld}$ tells Course in which a proper subset of female students are enrolled.

Q.14 (d)

Execute the given queries on the given tables. [key= PQ]

Options I, III, and IV are identical.

R	
P	Q
1	A
2	B
3	D

S	
P	Q
1	A
2	C
4	D

P
1
P
1
2

I) $\Pi_{\mathbf{P}}(R \bowtie S)$

II) Only II)$\Pi_{\mathbf{P}}(R) \bowtie \Pi_{\mathbf{P}}(S)$ returns

III) $\Pi_{\mathbf{P}}(\Pi_{\mathbf{P,Q}}(R) \wedge \Pi_{\mathbf{P,Q}}(S))$

IV) $\Pi_{\mathbf{P}}(\Pi_{\mathbf{P,Q}}(R) - (\Pi_{\mathbf{P,Q}}(R) - \Pi_{\mathbf{P,Q}}(S)))$ returns

Q.15 (c)

This is very clear from the options itself that the option 2 and option 4 are equivalent as option 2 is technical representation of code given in option 4. Therefore,

$\{t \mid t \in \pi_{R-S}(r) \in s(\exists v \in r(u = v[s] \wedge t = v[R-S]))\}$

= Select R.a, R.b

From R, S

= Where R.c = S.c

Q.16 (a)

Supplier sells other than blue part it can be red or green or both.

Q.17 (b)

Suppliers (Sid, Sname, City, Street)

1. Each supplier and each street within a city has a unique name Sid Street City Sname
2. (Sname, city) forms candidate key so that (Sname city) -+ Sid Street 3. Sid primary key so that Sid -> Sname City Street Each FD of above satisfy BCNF.

(a) RI (A, B) R2 (C, D) B is a foreign key and referring to C & C is candidate key So, IIB(ri) -11c(r2) = E E.g. B2

Q.18 (a)

Suppose, there are two relation schemas.

R_1 (A, B) and R_2 (C, D).

r_1 and r_2 are Two relation instances of R_1 and R_2 respectively. : -

B is a foreign key that refers to C in R_2.

B → Child Column & C → Parent Column

Child Column – Parent Column is always empty.

$\pi_B(r1)- \pi_C(r2)$ gives empty relation.

Q.19 (a)

"Find the distinct names of all students who score more than 90% in the course numbered 107".

1. **SQL Query Condition** would give all sname having score > 90 and attending course no. 107 and DISTINCT S.sname will give distinct student names, true
2. **Relational Algebra** π_{sname} gives projection of all students meeting the condition and π gives DISTINCT value, true
3. **Tuple Calculus** Gives DISTINCT student name having score > 90 and course No is 107, true
4. **Domain Calculus** Domain Calculus is equivalent to relational algebra and provides distinct value for the query, true.

Hence, all queries are equivalent to this query in English. (a) I, II, III and IV.

Q.20 (a)

Join will have fewer number of disk block accesses if outer loop has smaller relation (r(R)).

Q.21 (d)

$\pi_{empId}(employee) - \pi_{empId}(employee \bowtie_{(empId=eId) \wedge (empAge \leq depAge)} dependent)$

Where,

i) $\pi_{empId}(employee \bowtie_{(empId=eId) \wedge (empAge \leq depAge)} dependent)$

Means

All employees whose age is less than or equal to that of all of his dependents.

ii) $\pi_{empId}(employee) - \pi_{empId}(employe e \bowtie_{(empId=eId) \wedge (empAge \leq depAge)} dependent)$.

Means

All employees whose age is greater than that of all of his dependents.

Q.22 (c)

R_1

A	B
1	5
3	7

R_2

A	C
1	7
4	9

$R = R_1 \times R_2$

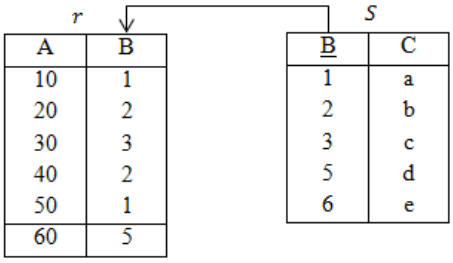
R contains e, f, g but not a, b.

R	A	B	C	
	1	5	7	→
	3	7	Null	→
	4	Null	9	→

Q.23 (c)

Consider the following relations r(A, B) and S(B, C), where S.B is a primary key and r.B is a foreign key

referencing S.B
Consider the following tables without NULL values.



A	B
10	1
20	2
30	3
40	2
50	1
60	5

B	C
1	a
2	b
3	c
5	d
6	e

Q: $r \bowtie (\sigma_{B<5}(S))$
The result of $\sigma_{B<5}(S)$ is

B	C
1	a
2	b
3	c

The result of $r \bowtie (\sigma_{B<5}(S))$ is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a

Option (A): The result of $r \bowtie S$ is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a
60	5	d

The result of $\sigma_{B<5}(r \bowtie S)$ is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a

Option (B): The result of r LOJ S is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a
NULL	6	e
NULL	5	d

The result of $\sigma_{B<5}(r \text{ LOJ } S)$ is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a

Option (C): The result of $\sigma_{B<5}(S)$ is

B	C
1	a
2	b
3	c

Now, the result of r LOJ($\sigma_{B<5}(S)$)

A	B
10	1
20	2
30	3
40	2
50	1
60	5

LOJ

B	C
1	a
2	b
3	c

=

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a
60	NULL	NULL

Option (D):
The $\sigma_{B<5}(r)$ is

A	B
10	1
20	2
30	3
40	2
50	1

Now, $\sigma_{B<5}(r)$ LOJ S is

A	B	C
10	1	a
20	2	b
30	3	c
40	2	b
50	1	a

Therefore, from the output of above four options, the results of options (a), (b) and (d) are equivalent to Q.

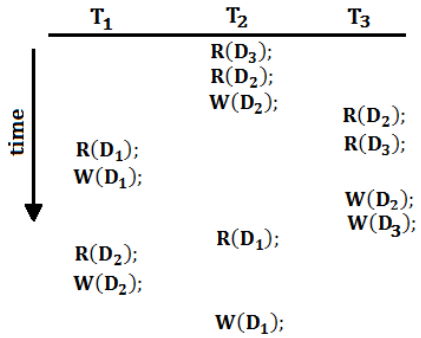
5 TRANSACTIONS & CONCURRENCY CONTROL

Q.1 Which of the following scenarios may lead to an irrecoverable error in a database system?

a) A transaction writes a data item after it is read by an uncommitted transaction
b) A transaction reads a data item after it is read by an uncommitted transaction
c) A transaction reads a data item after it is written by an uncommitted transaction
d) A transaction reads a data item after it is written by an uncommitted transaction

[GATE – 2003]

Q.2 Consider three data items D_1, D_2 and D_3 and the following execution schedule of transactions T_1, T_2 and T_3. In the diagram, R (D) and W (D) denote the actions reading and writing the data item D respectively.



Which of the following statements is correct?

a) The schedule is serializable as T_2; T_3; T_1
b) The schedule is serializable as T_2; T_1; T_3
c) The schedule is serializable as T_3; T_2; T_1
d) The schedule is not serializable

[GATE – 2003]

Q.3 Which level of locking provides the highest degree of concurrency in a relational database?

a) Page
b) Table
c) Row
d) Page, table and row level locking allow the same degree of concurrency

[GATE-2004]

Q.4 Consider the following schedule S of transactions Tl and T2:

T1	T2
Read(A) A = A-10	
	Read(A) Temp = 0.2 * A Write(A) Read(B)
Write(A) Read(B) B=B+10 Write(B)	 B= B+ Temp Write(B)

Which of the following is TRUE about the schedule S ?

a) S is serializable only as T1, T2
b) S is serializable only as T2, T1
c) S is serializable both as T1, T2 and T2, T1
d) S is not serializable either as Ti or as T2

[GATE-2004]

Q.5 Amongst the ACID properties of a transaction, the 'Durability' property requires that the changes made to the database by a successful transaction persist

a) Except in case of an Operating System crash
b) Except in case of a Disk crash
c) Except in case of a power failure
d) Always, even if there is a failure of any kind

[GATE-2005]

Q.6 Consider the following log sequence of two transactions on a bank

account, with initial balance 12000, that transfer 2000 to a mortgage payment and then apply a 5 % interest

1. T_1 start
2. T_1 B old = 1200 new = 10000
3. T_1 M old = 0 new = 2000
4. T_1 Commit
5. T_2 start
6. T_2 B old = 10000 new = 10500
7. T_2 Commit

Suppose the database system crashes just before tog record 7 is written. When the system is started, which one statement is true of the recovery procedure?

a) We must redo log record 6 to set)9 to 10500
b) We must undo log record 6 to set 13 to 10000 and then redo log records 2 and 3
c) We need not redo log records 2 and 3 because transaction T_1 has committed
d) We can apply redo and undo operations in arbitrary order because they are idempotent

[GATE – 2006]

Q.7 Consider the following schedules involving two transactions. Which one of the following statements is true?

S_1:$r_1(X)$;$r_1(Y)$; $r_2(X)$; $r_2(Y)$; $w_2(Y)$; $w_1(X)$

S_2:$r_1(X)$;$r_2(X)$;$r_2(Y)$; $w_2(Y)$; $r_1(Y)$;; $w_1(X)$

a) Both S1 and S2 are conflict serializable
b) S1 is conflict serializable and S2 is not conflict serializable
c) S1 is not conflict serializable and S2 is conflict serializable
d) Both Si and S2 are not conflict serializable

[GATE – 2007]

Q.8 Consider the following two transactions: T_1 and T_2.

T_1:	T2:
read(A);	read(B);
read(B);	read(A);
If A= 0 thenB←B+1;	If B 0 then A 4.- A←1;
write(B);	write(A);

Which of the following schemes, using shared and exclusive locks, satisfy the requirements for strict two phase locking for the above transactions?

a)

S1 :	**S2 :**
lock S(A);	lock S(B);
read (A);	read (B);
lock S(B);	lock S(A);
read (B);	read (A);
if A = 0	if B ≠ 0
then B ←B+1;	then A ←A-1;
write (B);	write (A);
commit;	commit;
unlock (A);	unlock (B);
unlock (B);	unlock (A);

b)

S1 :	**S2 :**
lock X(A);	lock X(B);
read (A);	read (B);
lock X(B);	lock X(A);
read (B);	read (A);
if A = 0	if B ≠ 0
then B← B+1;	then A ← A-1;
write (B);	write (A);
unlock (A);	unlock (A);
commit;	commit;
unlock (B);	unlock (A);

c)

S1 :	**S2:**
lock S(A);	lock S(B);
read (A);	read (B);
lock X(B);	lock X(A);
read (B);	read (A);
if A = 0	if B ≠ 0
then B← B+1;	then A ← A- 1;
write (B);	write (A);
unlock (A);	unlock (A);
commit;	commit;
unlock (B);	unlock (A);

d)

S1 :	**S2 :**
lock S(A);	lock S(B);
read (A);	read (B);
lock X(B);	lock X(A);
read (B);	read (A);
if A = 0	if B ≠ 0

then B ←B+1;	then A ←A-1;
write (B);	write (A);
unlock (A);	unlock (A);
unlock (B);	unlock (B);
commit;	commit;

[GATE-2007]

Q.9 Consider the following three schedules of transactions T1, T2 and T3. [Notation: In the following NYO represents the action Y (R for read, W for write) performed by transaction N on object 0].

S1:2RA 2WA 3RC 2WB 3WA 3WC 1RA 1RB 1WA 1WB

S2: 3RC 2RA 2WA 2WB 3WA 1RA 1RB 1WA 1WB 3WC

S3: 2RA 3RC 3WA 2WA 2WB 3WC 1RA 1RB 1WA 1WB

Which of the following statements is TRUE?

a) S1, S2 and S3 are all conflict equivalent to each other No two of S1,
b) S2 and S3 are conflict equivalent to each other
c) S2 is conflict equivalent to S3, but not to S1
d) S1 is conflict equivalent to S2, but not to S3

[GATE-2008]

Q.10 Consider two transactions T_1, and T_2, and four schedules S_1, S_2, S_3, S_4 of T_1 and T_2 as given below:

T_1: $R_1[x]$ W_1 [x] $W_1[y]$

T_2: $R_2[x]$ R_2 [y] W_2 [y]

S_1: $R_1[x]$ R_2 [x] R_2 [y] W_1 [x] $W_1[y]$ W_2 [y]

S_2: $R_1[x]$ R_2 [x] R_2 [y] W_1 [x] W_2 [y] $W_1[y]$

S_3: $R_1[x]$ W_1 [x]] R_2 [x] $W_1[y]$ R_2 [y] W_2 [y]

S_4: $R_2[x]$ R_2 [y] $R_1[x]$ W_1 [x] $W_1[y]$ W_2 [y]

Which of the above schedules are conflict-serializable?

a) S_1 and S_2 b) S_2 and S_3
c) S_3 only d) S_4 only

[GATE – 2009]

Q.11 Consider the following schedules for transactions T_1, T_2 and T_3

T_1	T_2	T_3
Read (X)		
	Read (Y)	
		Read (Y)
	Write (Y)	
Write (X)		
		Write (X)
	Read (X)	
	Write (X)	

Which one of the schedules below is the correct serialization of the above?

a) $T_1 \rightarrow T_3 \rightarrow T_2$ b) $T_2 \rightarrow T_1 \rightarrow T_3$
c) $T_2 \rightarrow T_3 \rightarrow T_1$ d) $T_3 \rightarrow T_1 \rightarrow T_2$

[GATE – 2010]

Q.12 Which of time following concurrency control protocols ensure both conflict serializability and freedom from deadlock?

1. 2-phase locking
2. Time-stamp ordering

a) 1 only b) 2 only
c) Both 1 and 2 d) neither 1 nor 2

[GATE – 2010]

Q.13 Consider the following transactions with data items P and Q initialized to zero:

T1	T2
read (P): read (Q) if P=0 then Q: Q+1 write (Q)	read (Q): read (P) if Q=0 then P: P+1 write (P)

Any non-serial interleaving of T_1 and T_2 for concurrent execution leads to

a) a serializable schedule
b) a schedule that is not conflict serializable
c) a conflict serializable schedule
d) a schedule for which a precedence graph cannot be drawn

[GATE – 2012]

Q.14 An index is clustered, if

a) it is on a set of fields that form a candidate key.
b) it is on a set of fields that include the primary key.
c) the data record of the file are organized in the same order as the date entries of the index
d) the data records of the file are organized not in the same order as the data entries of the index.

[GATE-2013]

Q.15 Consider the following four schedules due to three transactions (indicated by the subscript) using read and write on a data item x, denoted by r(x) and w(x) respectively. Which one of them is conflict serializable?
a) $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$
b) $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x)$
c) $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x)$
d) $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$

[GATE-2014]

Q.16 Consider the following schedule S of transactions T1, T2, T3 and T4:

T1	T2	T3	T4
Writes(X) Commit	Reads(X) Writes(Y) Writes(Y) Reads(Z) Commit	Writes(X) Commit	Reads(X) Reads(Y) Commit

Which one of the following statements is CORRECT?
a) S is conflict-serializable but not recoverable
b) S is not conflict-serializable but is recoverable .
c) S is both conflict-serializable and recoverable
d) S is neither conflict-serializable nor is it recoverable

[GATE-2014]

Q.17 Consider the transactions T1, T2, and T3 and the schedules S1 and S2 given below.

T1: rl (X); H. (Z); wl (X); wl(Z)
T2: r2 (Y); r2 (Z); w2 (Z)
T3: r3 (Y); r3 (.X); w3 (Y)
S_1 : rl (X); r3 (Y); r3 (X); r2 (Y); r2 (Z); w3(Y); w2 (Z); rl(Z); wl(X); wl(Z)
S_2 : r1 (X); r3 (Y); r2 (Y); r3 (X); ri(Z); r2(2); w3 (17); wl (X); w2 (2); wl(Z)
Which one of the following statements about the schedules is TRUE?
a) Only S1 is conflict-serializable.
b) Only S2 is conflict-serializable.
c) Both Si and S2 are conflict-serializable.
d) Neither S1 nor S2 is conflict-serializable.

[GATE-2014]

Q.18 Consider the following transaction involving two bank account x and y.read (x) ;
x : = x – 50;
write (x) ;
read (y);
y : = y + 50 ;
write (y);
The constraint that the sum of the accounts x and y should remain constant is that of
a) Atomicity b) Consistency
c) Isolation d) Durability

[GATE 2015]

Q.19 Consider a simple check pointing protocol and the following set of operations in the log. (Start, T4); (write, T4, y, 2, 3); (Start, T1); (commit, T4); (Write, T1, z, 5, 7); (checkpoint); (Start, T2); (write, T2, x, 1, 9); (commit, T2); (start, T3), (write, T3, z, 7, 2);
If a crash happens now and the system tries to recover using both undo and redo operations, what are the contents of the undo lists and the redo list?
a) Undo T3,T1; Redo T2
b) Undo T3,T1; Redo T2,T4

c) Undo: none; redo :T2,T4,T3,T1
d) Undo T3,T1; T4; Redo :T2

[GATE 2015]

Q.20 Consider the following partial Schedule S involving two transactions T1 and T2. Only the read and write operations have been shown. The read operation on data item P is denoted by read (P) and the write operation on data item P is denoted by write (P).

Time Instance	Transaction ID	
	T1	T2
1	Read(A)	
2	Write(A)	
3		Read(C)
4		Write(C)
5		Read(B)
6		Write(B)
7		Read(A)
8		Commit
9	Read(B)	

Suppose that the transaction T1 fails immediately after time instance 9.
Which one of the following statements is correct?

a) T2 must be aborted and then both T1 and T2 must be re-started to ensure transaction atomicity
b) Schedule S is non-recoverable and cannot ensure transaction atomicity
c) Only T2 must be aborted and then re-started to ensure transaction atomicity
d) Schedule S is recoverable and can ensure atomicity and nothing else needs to be done

[GATE 2015]

Q.21 Which one of the following is NOT a part of the ACID properties of database transactions?

a) Atomicity b) Consistency
c) Isolation d) Deadlock-freedom

[GATE 2016]

Q.22 Consider the following two phase locking protocol. Suppose a transaction T accesses (for read or write operations), a certain set of objects {O1,.....,Ok}. This is done in the following manner:
Step: 1 → "T" acquires exclusive locks to O1,...., Ok in increasing order of their addresses.
Step: 2 → The required operations are performed.
Step: 3 → All locks are released.
This protocol will

a) Guarantee Serializability and deadlock-freedom
b) Guarantee neither Serializability nor deadlock-freedom
c) Guarantee Serializability but not deadlock-freedom
d) Guarantee deadlock-freedom but not Serializability

[GATE 2016]

Q.23 Suppose a database schedule S involves transactions T1, ..., Tn. Construct the precedence graph of S with vertices representing the transactions and edges representing the conflicts. If S is Serializable, which one of the following orderings of the vertices of the precedence graph is guaranteed to yield a serial schedule?

a) Topological order
b) Depth-first order
c) Breadth-first order
d) Ascending order of transaction indices

[GATE 2016]

Q.24 Consider the following database schedule with two transactions T1 and T2.
S = r2(X); r1(X); r2(Y); w1(X); r1(Y); w2(X); a1; a2
Where "ri(Z)" denotes a read operation by transaction "Ti" on a variable Z, "wi(Z)" denotes a write operation by "Ti" on a variable Z and "ai" denotes an abort by transaction Ti. Which one of the following

statements about the above schedule is TRUE?

a) S is non-recoverable
b) S is recoverable, but has a cascading abort
c) S does not have a cascading abort
d) S is strict.

[GATE 2016]

Q.25 Consider a database that has the relation schema CR (Schema CR (Student name, Course Name). An instance of the schema CR is as given below:

CR

Student Name	Course Name
SA	CA
SA	CB
SA	CC
SB	CB
SB	CC
SC	CA
SC	CB
SC	CC
SD	CA
SD	CB
SD	CC
SD	CD
SE	CD
SE	CA
SE	CB
SF	CA
SF	CB
SF	CC

The following query is mad on the database.

$T_1 \leftarrow \pi_{CourseName}\left(\sigma_{StudentName='SA'}(CR)\right)$

$T_2 \rightarrow CR \div T_1$

The number of rows in T_2 is ___.

[GATE 2017]

Q.26 Consider a database that has the relation schemas EMP (EmpId, EmpName, Deptid) and DEPT (Dept name, Deptid). Note that the Deptid can be permitted to be NULL in the relation EMP. Consider the following queries on the database expressed in tuple relational calculus.

I $\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \wedge \forall v \in DEPT(t[DeptId] \neq v[DeptId]))\}$

II. $\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \exists \wedge v \in DEPT(t[DeptId] \neq v[DeptId]))\}$

III. $\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \wedge \exists v \in DEPT(t[DeptId] = v[DeptId]))\}$

Which of the above queries are safe?

a) I and II only b) I and III only
c) II and III only d) I, II and III

[GATE 2017]

Q.27 In a database system, unique timestamps are assigned to each transaction using Lamport's logical clock. Let TS(T_1) and TS(T_2) be the timestamps of transactions T_1 and T_2 respectively. Besides T_1 holds a lock on the resource R and T_1 has requested a conflicting lock on the same resource R. The following algorithm is used to prevent deadlocks in the database system assuming that a killed transaction is restarted with the same timestamp.

If TS(T_2) < TS(T_1) then

T_1 is killed

else T_2 waits

Assume an transaction that is not killed terminates eventually. Which of the following is TRUE about the database system that uses the above algorithm to prevent deadlocks?

a) The database system is both deadlock- free and starvation – free
b) The database system is deadlock free, but not starvation – free
c) The database system is starvation free, but not deadlock- free
d) The database system is neither deadlock –free nor starvation – free.

[GATE 2017]

Q.28 Two transactions T_1 and T_2 are given as

$T_1 : r_1(X)w_1(X)r_1(Y)w_1(Y)$

$T_2 : r_2(Y)w_2(Y)r_2(Z)w_2(Z)$

Where $r_i(V)$ denotes a read operation by transaction T_i on a variable V and $w_i(V)$ denotes a write operation by transaction T_i on a variable V. The total number of conflict serializable schedules that can be formed by T_1 and T_2 is _____.

[GATE 2017]

ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(d)	(d)	(d)	(d)	(c)	(c)	(c)	(c)	(c)	(b)	(a)	(b)	(b)	(c)	(d)
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**		
(c)	(a)	(b)	(a)	(b)	(d)	(a)	(a)	(c)	4	(d)	(a)	54		

EXPLANATIONS

Q.1 **(d)**
A transaction reads a data item after it is written by an uncommitted transaction. As the errors cannot recover in this situation.

Q.2 **(d)**
The given schedule is neither conflict serializable [cycle formed between T1 & T2]nor view serializable [for D1 order is T1→T2, and for D2 order is T2→T1], hence it is not serializable because as operations are not given we can't verify for result equivalence.

Q.3 **(d)**
A transaction reads a data item after it is written then we can't recover the errors in this situation by an uncommitted transaction.

Q.4 **(d)**
We have sequence W2(D2) →W1(D1)
Which means T2→T1 whereas we also have Wl(D1)→R2(D1) which means Ti T2. Both are not possible. Therefore, schedule is not serializable.

Q.5 **(c)**
Row level locking provides more concurrency. Because different transactions can access different rows in a table/page at same time.

Q.6 **(c)**
As it is given that the transaction is already committed, the changes are already made. Now, since the changes are made, they are permanent so we do need to restore them as the failure has no effect on the transaction.
We need not redo log records 2 and 3 because transaction T_1 has committed.

Q.7 **(c)**
Let's construct the S_1 and S_2 with the time to check the serializable conflict.

Time →

$S_1: r_1(X); r_1(Y) r_2(X); r_2(Y); w_2(Y); w_1(X)$
$S_2: r_1(X); r_2(X) r_2(Y); w_2(Y); r_1(Y); w_1(X)$

S_1 → No serializable conflict
S_2 → Serializable conflict in $W_1(x)$

Q.8 **(c)**
Requirement to follow Strict 2PL
1. Exclusive locks should be released after the commit .
2. No Locking can be done after the first Unlock and vice versa.
   In 2PL, deadlock may occur but it may be that it doesn't occur at all. Consider that in option (c) if both execute in serial order without concurrency then that is perfectly valid and YES it follows Strict 2PL.

Q.9 **(c)**
(i) Schedule S1: 2RA, 2WA, 3RC, 2WB, 3WA.. 3WC, 1RA, 1RB, 1WA, 1WB.

T1	T2	T3
	R(A)	
	W(A)	
		R(C)
	W(B)	
		W(A)
		W(C)
R(A)		
R(B)		
W(A)		
W(B)		

S1 is conflict serializable.
T2 →T3 → T1

(ii) Schedule S2: 3RC, 2RA, 2WA, 2WB, 3WA_ 1RA, 1RB, 1WA, 1WB, 3WC.

T1	T2	T3
		R(C)
	R(A)	
	W(A)	
	W(B)	
		W(A)
R(A)		
R(B)		
W(A)		
W(B)		
		W(C)

S2 is conflict serializable

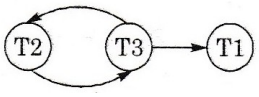
T2→T3→T1

S1 is conflict equivalent to S2.

iii) Schedule S3: 2RA, 3RC, 3WA, 2WA, 3WB, 3WC,1RA, 1RB,1WA, 1WB.

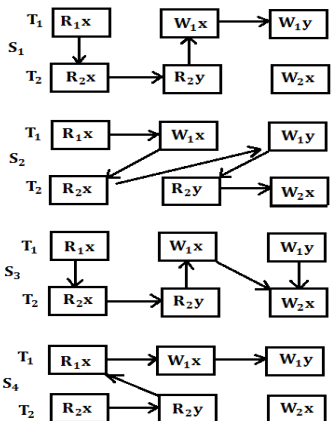
T1	T2	T3
	R(A)	
		R(C)
		W(A)
	W(A)	
	W(B)	
		W(C)
R(A)		
R(B)		
W(A)		
W(B)		

S3 is not conflict serializable.



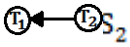
Therefore S1 and S2 are conflict cycle equivalent schedules but S3 is not equivalent to Si and S3.

Q.10 (b)



Let us construct the schedule and transition diagram to determine the conflict: Schedule S_2

T_1	T_2
R[X]	
	R[X]
	R[Y]
W[X]	
	W[Y]
W[Y]	

Dependency graph  have no cycles.

S_2

Schedule S_3

T_1	T_2
R[X]	
W[X]	
	R[X]
W[Y]	
	W[Y]
	W[Y]

Here, we can see that 5, and S_4 are not conflict serializable but S_2 and 5_3 are conflict serializable.

Q.11 (a)

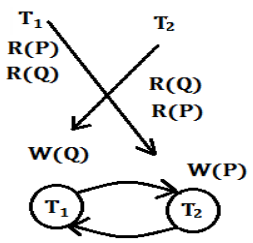
The solution can be obtained by checking with all the available options. The correct serialization is $T_1 \rightarrow T_3 \rightarrow T_2$.

This will result in same result as in the given schedule and will not conflict in read write operation of same data items.

Q.12 (b)

Time stamp ordering ensures both conflict serializability and freedom from deadlock as timestamp-based concurrency control is a non-lock concurrency control method. The method is employed in relational databases to safely handle transactions. It uses the timestamps for the same.

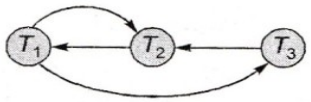
Q.13 (b)



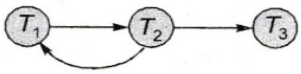
Precedence graph for the schedule says that it is not the conflict serializable schedule so option (c) is not correct.
Also option (d) is eliminated because graph can be drawn but it has cycle.
Above schedule is not even view serializable schedule (it fails the condition of view serializable schedule).
So, above schedule is not serializable schedule.

Q.14 (c)
Clustered index sort the data in the table based on their key values of the column on which clustered index is created. So option (c) is correct.

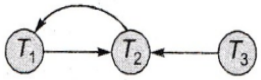
Q.15 (d)
(a) $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$



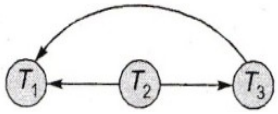
Contains cycle; Not conflict serializable
(b) $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x)$



Contains cycle; Not conflict serializable
(c) $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x)$

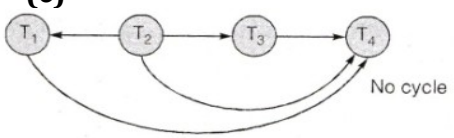


Contains cycle; Not conflict serializable
(d) $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$



Not contains cycle; conflict serializable.

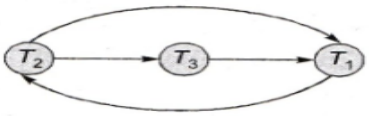
Q.16 (c)



∴ Conflict serializable and it is also recoverable.

Q.17 (a)
S1: r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); rl(Z); w1(X); w1(Z)

No cycle⇒ S1 is conflict serializable.
S2: r1(X); r3(Y); r2(Y); r3(X); rl(Z); r2(Z); w3(1); wl(X); w2(Z); w1(2)



Cycle S2 is not conflict serializable.

Q.18 (b)
Consistency in database systems refers to the requirement that any given database transaction must only change affected data in allowed ways, that is sum of x and y must not change.

Q.19 (a)
Since T1 and T3 are not committed yet, they must be undone. The transaction T2 must be redone because it is after the latest checkpoint.

Q.20 (b)
T2 reads value of 'A' which is written by T1 and T2 is committed before T1.

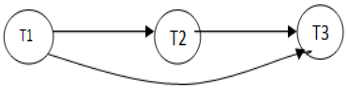
Q.21 (d)
Deadlock-freedom is not an ACID property of Transaction.
ACID Properties of transaction are
A → Atomicity
C → Consistency
I → Isolation
D → Durability

Q.22 (a)
Two Phase Locking Protocol → Always guarantee Serializability.

As Locks are requested in increasing order → No cycle will be formed, hence it is a deadlock free protocol.
The given technique guarantees Serializability & Deadlock Freedom

Q.23 (a)

Arranging vertices in Topological order guarantees the precedence graph yield serial schedule because, the arrangement of Topological order confirms that there is no cycle in the graph [Since Topological ordering is possible only for Directed Acyclic graph]. When there is no cycle, the corresponding schedule is Serializable and the ordering of vertices gives the resultant serial schedule [Linear arrangement of vertices based on paranthesis theorem].
Consider the below Topological ordering of vertices [transactions] which says that the resulting serial schedule is T1→T2→T3



Q.24 (c)

The given schedule does not have cascading aborts, since all Read Operations are committed Read only. To verify for recoverability and cascading aborts, we have to focus on WRITE-READ sequencing, we have to search for Dirty Read operation by any transaction.
Given Schedule → Has No Dirty Reads → Schedule has no cascading aborts.

T1	T2
	R2(X)
R1(X)	R2(Y)
W1(X)	
R1(Y)	W2(X)
Abort	
	Abort

Q.25 (4)

T_1 result :

Course Name
CA
CB
CC

T_2 result: $CR \div T_1 \Rightarrow$ student name for which every course name of CA, CB,CC is

SA
SC
SD
SF

4 tuples in result

Q26 (d)

I.

$$\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \wedge \forall v \in DEPT(t[DeptId] \neq v[DeptId]))\}$$

Results empname who does not belongs to any departments (safe query).

II.

$$\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \exists \wedge v \in DEPT(t[DeptId] \neq v[DeptId]))\}$$

Results empname who does not belongs to some departments (safe query).

II

$$\{t \mid \exists u \in EMP(t[EmpName] = u[EmpName] \wedge \exists v \in DEPT(t[DeptId] = v[DeptId]))\}$$

Results empname who belongs to some departments (safe query).
∴ All are safe queries.

Q.27 (a)

T_1 holds lock on resource R,
T_2 requires conflict lock on same resource R

Wait for graph
If $TS(T_2) < TS(T_1)$
The T_1 killed [restart with same TS value] else T_2 waits. Avoids both deadlock is starvation

Q.28 (54)

$T_1 : r_1(X)w_1(X)r_1(Y)w_1(Y)$

$T_2 : r_2(Y)w_2(Y)r_2(Z)w_2(Z)$

(i) Number of conflict serializable on

$T_1 \rightarrow T_2 : 1$

$r_1(X)w_1(X)r_1(Y)w_1(Y)r_2(Y)w_2(Y)r_2(Z)w_2(Z)$

(ii) Number of conflict seralizable o

$T_2 \rightarrow T_1 : 53$

$S : r_2(Y)w_2(Y)r_1(Y)w_1(Y)$

$r_1(X)w_1(X)$ must be before $r_1(Y)$

So that $(r_2(Y)w_2(Y))\,(r_1(Y)w_1(Y))$ can place.

${}^4C_2 = 6 \text{ways}$

1. $r_2(Y)w_2(Y)r_1(X)w_1(X)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place i ${}^6C_2 = 15 \text{ways}$

2. $r_2(Y)r_1(X)w_1(X)w_2(Y)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place ${}^4C_2 = 6 \text{ways}$

3. $r_2(Y)r_1(X)w_2(Y)w_1(X)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place i ${}^5C_2 = 10 \text{ways}$

4. $r_1(X)w_1(X)r_2(Y)w_2(Y)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place in ${}^4C_2 = 6 \text{ways}$

5. $r_1(X)r_2(Y)w_2(Y)w_1(X)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place i ${}^5C_2 = 10 \text{ways}$

6. $r_1(X)r_2(Y)w_1(X)w_2(Y)r_1(Y)w_1(Y)$

   $r_2(Z)w_2(Z)$ can place in .

Total conflict serializable of T_1 and T_2 =53+1=54 ways

6 FILE STRUCTURES

Q.1 B^+ trees are preferred to binary trees in databases because
a) Disk capacities are greater than memory capacities
b) Disk access is much slower than memory access
c) Disk data transfer rates are much less than memory data transfer rates
d) Disks are more reliable than memory

[GATE – 2000]

Q.2 A B^+ tree index is to be built on the Name attribute of the relation STUDENT. Assume that all student names are of length 8 byte, disk blocks are of size .512- byte and index pointers are of size 4 byte. Given this scenario, what would be the best choice of the degree (i.e, the number of pointers per node) of the B^+ tree?
a) 16 b) 42
c) 43 d) 44

[GATE – 2002]

Q.3 The order of an internal node in a B^+ tree index is the maximum number of children it can have. Suppose that a child pointer takes 6 byte, the search field value takes 14 byte, and the block size is 512 byte. What is the order of the internal node?
a) 24 b) 25
c) 26 d) 27

[GATE – 2004]

Q.4 Consider a table T in a relational database with a key field K. A B – tree of order p is used as an access structure an K where p denotes the maximum number of tree pointers in a B-tree index node. Assume that K is 10 bytes long; disk block sixe is 512bytes; each data pointer P_D is 8 bytes long and each block pointer P_B is 5 bytes long .In order for each B-tree node to fit in a single disk block, the maximum value of p is
a) 20 b) 22
c) 23 d) 32

[GATE-2004]

Q.5 Which one of the following is a key factor for preferring B^+ trees to binary search trees for indexing database relations?
a) Database relations have a large number of records
b) Database relations are sorted on the primary key
c) B^+ trees require less memory than binary search trees
d) Data transfer from disks is in blocks

[GATE – 2005]

Q.6 A.B Tree used as an index for a large database table has four levels including the root node. If a new key is inserted in this index, then the maximum number of nodes that could be newly created in a the process are
a) 5 b) 4
c) 3 d) 2

[GATE-2004]

Linked Answer Questions Q.7& Q.8

A database table T1 has 2000 records and occupies 80 disk blocks. Another table T2 has 400 records and occupies20 disk blocks. These two tables have to be joined as per a specified join condition that needs to be evaluated for every pair of records from these two tables. The memory buffer space available can hold exactly one block of records for T1 and one block of records for T2 simultaneously at any point in time. No index is available on either table.

Q.7 If Nested – loop join algorithm is employed to perform the join, with the most appropriate choice of table to be used in outer loop, the number of block accesses required for reading the data are

a) 800000 b) 40080
c) 32020 d) 100

[GATE-2005]

Q.8 If, instead of Nested – loop join, Block nested – loop join is used, again with the most appropriate choice of table in the outer loop, the reduction in number of block accesses required for reading the data will be

a) 0 b) 3040
c) 38400 d) 798400

[GATE-2005]

Q.9 In a database file structure , the search key field is 9 bytes long, block size is 512 bytes, a record pointer is 7 bytes and a block pointer is 6 bytes .The largest possible order of a non- leaf node in a B + tree implementing this file structure is

a) 23 b)24
c) 34 d) 44

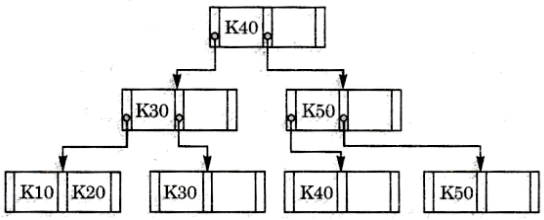
[GATE-2005]

Q.10 The order of a leaf node in a B* tree is the maximum number of (value, data record pointer) pairs it can hold. Given that the block size is 1 Kbyte, data record pointer is 7 byte long, the value field is 9 byte long and a block pointer is 6 Byte long, what is the order of the leaf node?

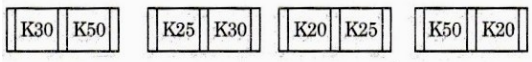
a) 63 b) 64
c) 67 d) 68

[GATE – 2007]

Directions for question Q.11to Q.12:
Consider the B+ tree in the adjoining figure, Where each node has at most two keys and three links.



Q.11 Keys K15 and then K25 are inserted into this tree in that order. Exactly how many of the following nodes (disregarding the links) will be present in the tree after the two insertions?



a)1 b)2
c)3 d)4

[GATE-2007]

Q.12 Now the key K50 is deleted from the B+ tree resulting after the two insertions made earlier. Consider the following statements about the B+ tree resulting after this deletion.

i) The height of the tree remains the same.
ii) The node k20 (disregarding the links) is present in the tree.
iii) The root node remains unchanged (disregarding the links)Which one of the following options is true?

a) Statements (i)and (ii) are true
b) Statements (ii) and (iii) are true
c) Statements (iii) and (i) are true
d) All the statements are false

[GATE-2007]

Q.13 Consider a file of 16384 records. Each record is 32 bytes long and its key field is of size 6 bytes. The file is ordered on a non-key field, and the file organization is unspanned. The file is stored in a file system with block size 1024 bytes, and the size of a block pointer is 10 bytes. If the secondary index is built on the key field of the file, and multi-level index scheme is used to store the

secondary index, the number of first-level and second-level blocks in the multi-level index are respectively?

a) 8 and 0 b) 128 and 6
c) 256 and 4 d) 512 and 5

[GATE-2008]

Q.14 A clustering index is defined on the fields which are of type

a) non-key and ordering,
b) non-key and non-ordering
c) Key and ordering
d) key and non-ordering

[GATE – 2008]

Q.15 The following key values are inserted into a B^+ tree in which order of the internal nodes is 3, and that of the leaf nodes is 2, in the sequence given below. The order of internal nodes is the maximum number of tree pointers in each node, and the order of leaf nodes is the maximum number of data items that can be stored in it The B^+tree is initially empty. 10, 3, 6,8,4,2, 1 The maximum number of times leaf nodes would get split up as a result of these insertions is

a) 2 b) 3
c) 4 d) 5

[GATE – 2009]

Q.16 Consider a B^+ tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node?

a) 1 b) 2
c) 3 d) 4

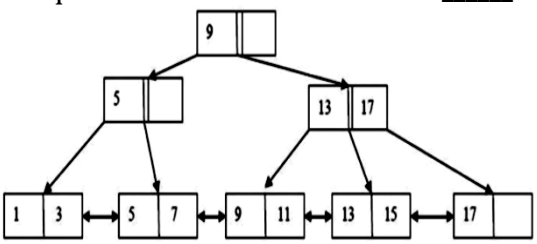
[GATE – 2010]

Q.17 A file is organized so that the ordering of data records is the same as or close to the ordering of data entries in some index. Then that index is called

a) Dense b) Sparse
c) Clustered d) Unclustered

[GATE-2015]

Q.18 With reference to the B+ tree index of order 1 shown below, the minimum number of nodes (including the Root node) that must be fetched in order to satisfy the following query: "Get all records with a search key greater than or equal to 7 and less than 15" is _____



[GATE 2015]

Q.19 Consider a B+ tree in which the search is 12 bytes long, block size is 1024 bytes, record pointer is 10 bytes long and block pointer is 8 bytes long. The maximum number of keys that can be accommodated in each non-leaf node of the tree is ___

[GATE-2015]

Q.20 B+ Trees are considered BALANCED because

a) the lengths of the paths from the root to all leaf nodes are all equal.
b) the lengths of the paths from the root to all leaf nodes differ from each other by at most 1.
c) the number of children of any two non-leaf sibling nodes differ by at most1.
d) the number of records in any two leaf nodes differ by at most1

[GATE-2016]

Q.21 In a B^+ tree, if the search-key value is 8 bytes long, the block size is 512 bytes and the block pointer size is 2 bytes, then the maximum order of the B^+ tree is_____.

[GATE-2017]

ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(b)	(c)	(c)	(c)	(d)	(a)	(c)	(b)	(c)	(a)	(a)	(a)	(c)	(a)	(a)
16	**17**	**18**	**19**	**20**	**21**									
(b)	(c)	5	50	(a)	2									

EXPLANATIONS

Q.1 **(b)**

We know that indexing is better if the data block is large.

Now, B+ trees are preferred over the binary search trees as in B+ trees, transfer of data is in form of data blocks. These data blocks can store large information while transferring; information on a single block is more efficient.

Q.2 **(c)**

Size of disc block =512

Size of index pointer = 4

4n+8(n-1) <= 512

12n <= 520

n<= 43.33

n must be an integer, so degree is 43.

Q.3 **(c)**

Let the order of the internal node be X,

Then from the given,

(X -1)14 + 6X <= 512

14X -14 + 6X <=512,

20X < 526

X =26

Q.4 **(c)**

$$(p-1)(key\_ptr\_size + record\_ptr\_size) + p\,(block\_ptr\_size) \le 512$$

Therefore p = 23

Q.5 **(d)**

We know that indexing is better if the data block is large.

Now, B+ trees are preferred over the binary search trees as in B+ trees, transfer of data from disk to primary memory is in form of data blocks. These data blocks can store large information while transferring; information on a single block is more efficient.

Q.6 **(a)**

Suppose all nodes are completely full that means every node has n-1 keys.

Tree has 4 levels. If a new key is inserted then at every level new node will be created.

In worst case root node will also be broken into two parts and we have 4 levels. So answer should be 5 because tree will be increased with one more level

Q.7 **(c)**

Nested Loop algorithm will involve $n_r * b_s + b_r$ block transfers.

Either T1 can be R or T2

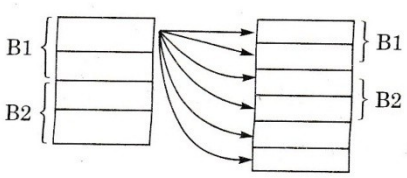
If r is T1 then total number of block access is 2000*20 + 80 = 40080

If R is T2 then total number of block access is 400*80 + 20 = 32020

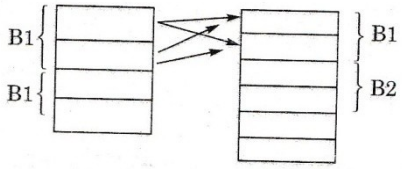
Better one is the second case, total number of block accesses = (32020)

Q.8 **(b)**

In nested loop join for each tuple in first table we scan through the entire tuples second table.



Here we will take table T2 as the outer table in nested loop join algorithm. The number of block accesses then will be 2
20+(400×80)=32020
In block nested loop join we keep 1 block of T1 in memory and 1 block of T2 in memory and do join on tuples.



For every block in T1 we need to load all blocks of T2. So number of block accesses is
80*20+1620
The difference is 32020-1620 = 30400

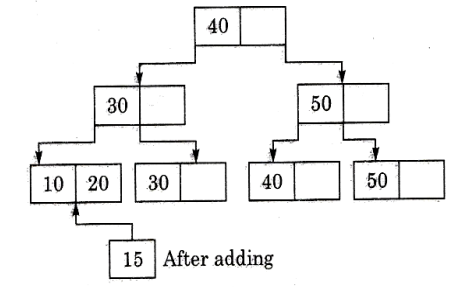
Q.9 (c)

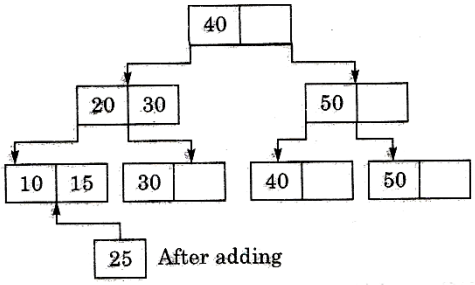
From the structure of B+ tree we can get the following equation:
n*p+(n-1)*(k + q) ≤ B
where n= order,
p = tree/ block/index pointer,
q= record/ data pointer,
B= size of block
In B+ tree, non leaf node has no record pointer:
(by putting q=0)
N*p+(n-1)k≤ B
n*6+(n-1)*9≤ 512
n≤34.77
largest possible order of a non- leaf node=34.

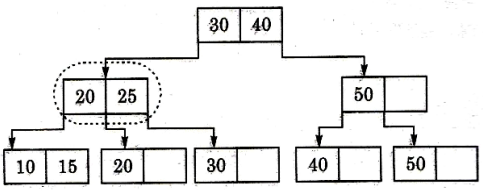
Q.10 (a)

Let the order of leaf node is n.
As per given, Block size = 1 k = 1024
$\Longrightarrow 6 + 7n + 9n = 1024$
$\Longrightarrow 16n = 1024 \Longrightarrow n = 63$

Q.11 (a)







Matching node is 20 25 which is one only.

Q.12 (a)

Only (i) and (ii) are correct. After deleting 50 from the tree we are left with node (20, 40) with 40 having no right subtree except 40 itself. Nodes can't combined because that would overflow the node as they are already half –full or full.
So key 40 can be out in node containing 30, Height remains same with 20 at root.

Q.13 (c)

Type of Index = Secondary Index (Key) → DENSE INDEX
Size of the Key = 6 bytes
Size of block pointer = 10 bytes
Size of the Block = 1024 bytes
Size of Index Record=10+6=16 bytes
No of Index Records that can fit in a single block =1024/16=64
Total No of Index Records = DENSE = 16384 index records
Total No of Blocks for First Level Index = 16384/64=**256** blocks

Total No of Blocks for Second Level Index = 256/64=**4** block

Q.14 (a)

We are given that the clustering index is defined on the fields. Now, if the records of the file are physically ordered on a non-key field since it is a non-key, it will not have a distinct value for each record.

Therefore, the clustering index is defined on the fields of type non-key and ordering.

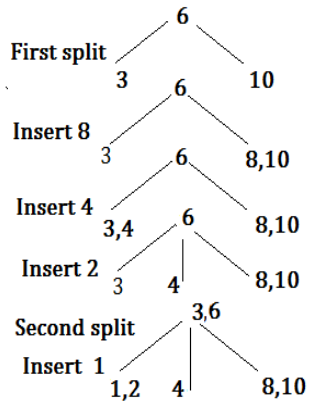
Q.15 (a)

No. of splits would be three. This can be further displayed in the figure below

Insert 10 : 10

Insert 3 : 3,10

Insert 6 :



Therefore, the total number of splits is 2.

Q.16 (b)

In B^+ tree, the root node has minimum 2 block pointers and maximum P block pointer where, P = order where, key=order -1 and in the non-root node, the minimum number of keys $= \left\lceil \frac{P}{2} \right\rceil - 1$

So, in the question key = 5, order = 6

So, minimum number of keys in non-root node $= \left\lceil \frac{6}{2} \right\rceil - 1 = 2$

Q.17 (c)

In Clustered Index, data blocks are stored in a way to match the index. Therefore, only one clustered index can be created on a given database table.

Q.18 (5)

We can get all values in range from 7 to 59 by accessing 5 nodes.

1) First search 7 in a leaf node.
2) Once 7 is found, linearly traverse till 15 is found.

Q.19 (50)

Let m be the order of B+ tree

$m(8)+(m-1)12 <= 1024$

$m <= 51$. Since maximum order is 51, maximum number of keys is 50.

Q.20 (a)

B+ Trees → Balanced Trees → All leaf nodes are at the same level.

Q.21 (2)

Let T_1 and T_2 be two registers,

$T_1 \leftarrow b$

$T_2 \leftarrow c$

$T_1 \leftarrow (T_{1+} T_2)$

$T_1 \leftarrow T_1/3$

$T_2 \leftarrow d$

$T_1 \leftarrow T_1 + T_2$

$T_2 \leftarrow a$

$T_2 \leftarrow T_2 - 1$

$T_1 \leftarrow T_1 * T_2$

Hence only 2 registers are required.

ASSIGNMENT QUESTIONS

Q.1 Which normal form is considered adequate for relational database design?
a) 2 NF b) 3 NF
c) 4 NF d) BCNF

Q.2 The concept of locking can be used to solve the problem of
a) Lost update
b) uncommitted dependency
c) Inconsistent data
d) deadlock

Q.3 Given relations R (w, x) and S(x, y), the result of
SELECT DISTINCT w, x
FROM R, S
Is guaranteed to be the same as R, if
a) R has no duplicates and S is non-empty
b) R and S have no duplicates
c) S has no duplicates and R is non-empty
d) R and S have the same number of tuples

Q.4 A functional dependency of the form $X \rightarrow Y$ is trivial if
a) $Y \subseteq X$ b) $Y \subset X$
c) $X \subseteq Y$ d) $X \subset Y$ and $Y \subset X$

Q.5 If every non-key attribute is functionally dependent on the primary key, then the relation will be in
a) First normal form
b) Second normal form
c) Third normal form
d) Fourth normal form

Q.6 The column of a table is referred to as the
a) tuple b) attribute
c) Entity d) degree

The next four questions are based on the following details. Consider the given schemes.
Branch_scheme = (Branch_name , assets , Branch_city)
Customer_scheme = (Customer_name , street , Customer_city)
Deposit_scheme = (Branch_name ,account_number, Customer_name,Balance)
Branch_scheme = (Branch_name, loan_number , Customer_name, amount)
Client_scheme = (Customer_name , Banker , banker_name)

Q.7 Using relational algebra, the query that finds customers who have a balance of over 1000 is
a) $\pi_{customer\_name}(\sigma_{balance}>1000(Deposit))$
b) $\sigma_{customer\_name}(\pi_{balance} > 1000\ (Deposit))$
c) $\pi_{customer\_name}(\sigma_{balance}> 1000\ (Borrow))$
d) $\sigma_{customer\_name}(\pi_{balance}> 1000\ (Borrow))$

Q.8 Which of the following queries finds the clients of banker Agassi and the city they live in?
a) πclient_customer_name,customer_city (σ client.customer_name = customer.customer_name (σ banker_name = “Agassi” (Client × Customer)))
b) π customer_name.customer_city(σ banker_name = “Agassi” (Client × Customer))
c) π client.customer_name.customer_city (σ banker_name=”Agassi”(σ client.customer_name=customer.customer_name (Client × Customer)))
d) π customer_name.customer_city (σ banker_name=”Agassi” (Client × Customer))

Q.9 Which of the following tuple relational calculus finds all customers who have a loan amount of more than 1200?

a) {t (Customer_name)| t∈borrow ∧ t [amount] > 1200}
b) {t | t(Customer_name) ∈ borrow ∧ t [amount] > 1200}
c) {t|∃s∈ borrow (t [Customer_name] = s [Customer_name] ∧ s [amount] > 1200)}
d) None of the above

Q.10 Which of the following Domain relational calculus finds all customers who have a loan amount of over 1200?
a) {<c>| ∃ b, 1, a (<b, 1, c, a >∈ borrow ∨a>1200)}
b) {<c>|∃ b, 1, a (<b, 1, c, a >∈ borrow ∧ a > 1200) }
c) {<c> | ∃<b, 1, c , a>∈ borrow ∧ a > 1200)}
d) {<c> | <b, 1, c, a>∈ borrow ∧ a > 1200)}

Q.11 Given the function dependencies $X \rightarrow W$; $X \rightarrow Y$; $Y \rightarrow Z$ and $Z \rightarrow PQ$ Which of the following does not hold good?
a) $X \rightarrow Z$ b) $W \rightarrow Z$
c) $X \rightarrow WY$ d) None of the above

Q.12 What are the potential problems when a DBMS executes multiple transactions concurrently?
a) The lost update problem
b) The dirty read problem
c) The unrepeatable read problem
d) The phantom problem

Q.13 The data flow model of an application mainly shows
a) The underlying data and the relationships among them
b) Processing requirements and the flow of data
c) Decision and control information
d) Communication network structure

Q.14 Consider the set of relations given below and the SQL query that follows:
Students:
(Roll_number, Name, Date_of_birth)
Courses:
(Course_number,Course_name, Instructor)
Grades:
(Roll_number, Cource_name, Grade)
SELECT DISTINCT Name
FROM Students, Courses, Grades
Where
Students.Roll_number=Grades.Roll_number
AND Courses. Instructor = Korth
AND Courses.Course_number = Grades. Course_number
AND Grades. Grade = A
Which of the following sets is computed by the above query?
a) Names of the students who have got an A grade in all courses taught by Korth
b) Names of the students who gave got an A grade in all courses
c) Names of the students who have got an A grade in at least one of the courses taught by Korth
d) None of the above

Q.15 Which of the following desired features are beyond the capability of relational algebra?
a) Aggregate computation
b) Multiplication
c) Finding transitive closure
d) None of the above

Q.16 In airline reservation system, the entities are date, flight number, place of departure, destination, type of plane and seats available. The primary key is
a) Flight number
b) Flight number + place of departure
c) Flight number + date
d) Flight number + destination

Q.17 For a database relation R(a, b, c, d) where the domains of a, b, c and d

include only atomic values, only the following functional dependencies and those can be inferred from them hold.

$a \rightarrow c$

$b \rightarrow d$

The relation is in

a) First normal form but not in second normal form
b) Second normal form but not in third normal form
c) Third normal form
d) None of the above

Q.18 Consider the following relation schema pertaining to a student database.

Student: (rollno, name, address)

Enroll: (rollno, courseno, course name)

Where the primary keys are shown underlined. The number of tuples in the student and Enroll tables are 120 and 8 respectively. What are the maximum and minimum number of tuples that can be present in (Student*Enroll), where '*' denotes natural join?

a) 8, 8 b) 120, 8
c) 960, 8 d) 960, 120

Q.19 In an E-R diagram ellipses represent

a) Entity sets
b) Relationship among entity sets
c) Attributes
d) Link between attributes and entity sets

Q.20 The set of permitted values for each attribute is called is

a) attribute set b) attribute range
c) domain d) group

Q.21 In an entity relationship, y is the domain of entity and x is a subordinate entity. Which of the following is/are incorrect?

a) Operationally, if y is deleted, so x is
b) x is existence, dependent on y
c) Operationally, x is deleted, so is y
d) Operationally, x is deleted, y remains the same

Q.22 A data model is a collection of conceptual tools for describing

a) Data and data relationship
b) Data semantics and consistency constraints
c) Data, data relationship, data semantics, consistency constraints
d) None of the above

Q.23 The employee information in a company is stored in the relation

Employee: (name, sex, salary, dept Name)

Assume name is the primary key. Consider the SQL query

```
Select      dept Name
Form        employee
WHERE       sex = 'M'
GROUP BY    dept Name
HAVING avg (salary) > (select avg (salary) FROM Employee
```

It returns the name of the departments which the average salary

a) Is more than the average salary in the company
b) Of the male employee is more than the average salary of all the male employees in the company.
c) Of the male employee is more than the average salary of the employee in the same department.
d) Of the male employee is more than the average salary in the company

Q.24 Choose the correct statements.

a) Relational algebra and relation calculus are both procedural query languages.
b) Relational algebra and relation calculus are both non-procedural query languages.
c) Relational algebra is a procedural query language and relation calculus is a non-procedural query languages.

d) Relational algebra is a non-procedural query language and relation calculus is a procedural query languages.

Q.25 Choose the incorrect statements.
a) In network model, data is represented by a collection of records and relationship among data are represented by links.
b) In hierarchical model, data and relationships among data are represented by records and links respectively.
c) In hierarchical model, the records are organized as a collection of arbitrary Graphs
d) In network model, the records are organized as a collection of trees

Q.26 Students and courses enrolled, is an example of
a) One-to-one relationship
b) one-to-many relationship
c) many-to-one relationship
d) many-to-many relationship

Q.27 Relations produced from an E-R model will always be in
a) First normal form
b) second normal form
c) Third normal form
d) fourth normal form

Q.28 Choose the correct statement
a) An alternate key is a candidate key that is not a primary key
b) An alternate key is a primary key that is not a candidate key
c) An alternate key is a candidate key that is also a primary key
d) None of the above

Q.29 E-R modelling technique is a
a) top-down approach
b) bottom-up approach
c) left-right approach
d) none of the above

Q.30 Third normal form is inadequate in situations where the relation
a) Has multiple candidate keys
b) Has candidate keys that are composite
c) Has overlapped candidate keys
d) None of the above

Q.31 Redundancy is dangerous as it is potential threat to data
a) Integrity b) consistency
c) Sufficiency d) none of the above

Q.32 Let R be a relation. Which of the following comments about the relation R are correct?
a) R will necessarily have a composite key if R is in BCNF but not in 4NF
b) If R is in 3NF and if every key of R is simple, then R is BCNF
c) If R is in BCNF and if R has at least one simple key, then R is 4NF
d) If R is in 3NF and if its every key is simple, then R is 5NF

Q.33 An attribute of one table matching the primary key of another table, is called as
a) Foreign key b) secondary key
c) Candidate key d) composite key

Q.34 A primary key if combined with a foreign key creates
a) Parent child relationship between the tables that connect them
b) many-to-many relationship between the tables that connect them
c) Network model between the tables that connect them
d) None of the above

Q.35 Choose the correct statement
a) Network models are complicated by physical keys, but the relation model is faster because it uses logical keys
b) Network models are complicated by logical keys, but the relation

model is faster because it uses physical keys
c) Network models are complicated by logical keys, but the relation model is slower because it uses physical keys
d) Network models are complicated by physical keys, but the relation model is slower because it uses logical keys

Q.36 The employee salary should not be greater than Rs. 2000. This is
a) Integrity constraint
b) referential constraint
c) over-defined constraint
d) feasible constraint

Q.37 Manager's salary details are hidden from the employee. This is
a) conceptual level data hiding
b) physical level data hiding
c) external level data hiding
d) none of the above

Q.38 The SQL expression
SELECT distinct T.branch_name from branch T, branch S Where T.assets>S.assets and S.assets and s.branch_city = "PONDICHERRY"
Finds the names of
a) all branches that have greater assets than any branch located in PONDICHERRY
b) all branches that have greater assets than all branch located in PONDICHERRY
c) all branches that has the greatest assets in PONDICHERRY
d) any branches that has greater assets than any branch located in PONDICHERRY

Q.39 A trigger is
a) A statement that enable to start any DBMS
b) A statement that is executed by the user when debugging an application program
c) A condition the system tests for the validity of the database user
d) A statement that is executed automatically by the system as a side effect of a modification to the database

The next two questions are based on the following information
Entity set TRANSACTION has the attributes transaction number, date, and amount
Entity set ACCOUNT has the attributes account number, customer name, balance

Q.40 Which is the discriminator of the weak entity?
a) Account number
b) Transaction number
c) {Account number, date}
d) date

Q.41 Which is the primary key of the weak entity?
a) Account number
b) {Account number, Transaction number}
c) {Account number, date}
d) {Transaction number, date}

Q.42 Consider the relation
EMPLOYEE (Emp-no, Emp-name, salary, project-no, due-date)
(Assuming an 1-1 relationship between project and employees)
Project-no is functionally dependent on
a) Emp-name b) Emp-no
c) due-date d) none of the above

Q.43 In the previous question, which of the following is/are functionally dependent on Emp-name? (Assume no two persons have the same name)
a) Emp-no
b) Project-no, salary
c) Salary, due date
d) Emp-no, salary, project-no, due date

Q.44 Let R(a, b, c) and S(d, e, f) be two relations in which d is the foreign key of S that refers to the primary key of R. Consider the following four operations.

a) Insert into R b) Insert into S
c) Delete from R d) Delete from S

Which of following is true about the referential integrity constraint above?

a) None of them can cause any violation
b) All of the can cause violation
c) Operations a) and b) can cause violation
d) Operations b) and c) can cause violation

Q.45 Which of the following schemes are used for ensuring atomicity?

a) Log with deferred modification
b) Log with immediate modification
c) Shadow paging
d) None of the above

Q.46 If P and Q are predicates and P is the relational algebra expression, then which of the following equivalence are valid?

a) $\sigma_P(\sigma_Q(e)) = \sigma_Q(\sigma_P(e))$_
b) $\sigma_P(\sigma_Q(e)) = \sigma_{P \wedge Q}(e)$
c) $\sigma_Q(\sigma_P(e)) = \sigma_{P \wedge Q}(e)$
d) None of the above

Q.47 Generally speaking, for a weak entity set to be meaningful it must be part of a

a) One-to-one relationship
b) One-to-many relationship
c) many-to-many relationship
d) None of the above

Q.48 If a relation scheme is in BCNF, then it is also in

a) First normal form
b) Second normal form
c) Third normal form
d) None of the above

Q.49 Consider the following set of functional dependencies on the scheme (A, B, C).

$A \rightarrow BC$
$B \rightarrow C$
$A \rightarrow B$
$AB \rightarrow C$

The canonical cover for this set is

a) $A \rightarrow BC$ and $B \rightarrow C$
b) $A \rightarrow BC$ and $AB \rightarrow C$
c) $A \rightarrow BC$ and $A \rightarrow B$
d) $A \rightarrow B$ and $B \rightarrow C$

Q.50 Assume transaction A holds a shared lock R. If transaction B also requests for a shared lock on R, it will

a) Result in a deadlock situation
b) Immediately be granted
c) Immediately be rejected
d) B granted as soon as it is released by A

Q.51 The index consists of

a) A list of keys
b) Pointers to the master list
c) Both a) and b)
d) All of the above

Q.52 An audit trail

a) Is used to make back-up copies
b) Is the recorded history of operations performed on a file ?
c) Can be used to restore lost information
d) All of the above

Q.53 A scheme describes

a) Data elements
b) Records and files
c) Record relationships
d) All of the above

Q.54 What is the abbreviation used for a software package that permits the users to create, retrieve and maintain records in a data base?

a) DASD b) FMS
c) EMMS d) DBMS

Q.55 In order to use a record management system

a) You need to understand the low level details of how information in stored

b) You need to understand the model the record management system uses
c) Both a) and b)
d) All of the above

Q.56 A form defines
a) Where data is placed on the screen
b) The width of each field
c) Both a) and b)
d) All of the above

Q.57 A top-to-bottom relationship among the items in a data base is established by
a) Hierarchical schema
b) Network schema
c) Relational schema
d) All of the above

Q.58 In a relational schema, each tipple is divided into fields called
a) Relations b) Domains
c) Queries d) All of the above

Q.59 A logical schema
a) Is the entire data base
b) Is a standard way of organizing information into accessible parts
c) Describes how data is actually stored on disk
d) All of the above

Q.60 Which of the following are not a relational data base?
a) DBASE IV b) 4th Dimension
c) FoxPro d) Reflex

Q.61 Which of the following contains a complete record of all activity that affected the contents of a data base during a certain period of time?
a) Report write
b) Query language
c) Data manipulation language
d) Transaction log

Q.62 When performing a look-up operation using a form
a) You enter the search value into the form
b) You look at each form sequentially until you see the one you want
c) You type the key in an entry line, and the correct form is displayed
d) All of the above

Q.63 What name is given to the collection of facts, items of information or data, which are relation in some way?
a) Data base
b) Directory information
c) Information tree
d) Information provider

Q.64 In any hierarchy of data organization, the smallest entity to be processed as a single unit called
a) Data field b) Data record
c) Data file d) Data base

Q.65 A network schema
a) Restricts the structure to a one-to-many relationship
b) Permits many to many relationships
c) Stores data in tables
d) All of the above

Q.66 A number of related records that are treated as a unit is called
a) File b) Field
c) Data d) Batch

Q.67 Which two files are used during operation of the DBMS?
a) Query language and utilities
b) Data manipulation language and query language
c) Data dictionary and transaction log
d) Data dictionary and query language

Q.68 A transparent DBMS

a) Cannot hide sensitive information from users
b) Keep its logical structure hidden from users
c) Keeps its physical structure hidden from users
d) Both b) and c)

Q.69 Related fields in a data base are grouped to form
a) Data file b) Data record
c) Menu d) Bank

Q.70 Goals for the design of the logical schema include
a) Avoiding data inconsistency
b) Being able to construct queries easily
c) Being able to access data efficiently
d) All of the above

Q.71 The designer of a form includes
a) Filed designators
b) Prompts
c) Data
d) None of the above.

Q.72 Batch processing is appropriate if
a) A large computer system is available
b) Only a small computer system is available
c) Only a few transactions are involved
d) All of the above

Q.73 A multiple-form file management system
a) Let's you define different forms for different operations
b) Lets you create a look-up form with an associated read-only password to prevent access by unauthorized users
c) Both a) and b)
d) Allow you to entry data in all forms

Q.74 The physical location of a record is determined by a mathematical formula that transforms a file key into a record location in
a) A tree file b) An indexed file
c) A hashed file d) A sequential file

Q.75 The relational model uses some unfamiliar terminology. A tuple is equivalent to a
a) Record b) Field
c) File d) Data base

Q.76 Which command (&) is (are) used to redefine a column
of the table in SQL ?
a) ALTER TABLE
b) DEFINE TABLE
c) MODIFY TABLE
d) None of the above.

Q.77 In SQL, the ALTER TABLESPACE Command is used
a) To add/rename data files
b) To change storage characteristics
c) To take a table space online/offline
d) To begin/end a backup.

Q.78 In SQL which command(s) is/are used to enable/disable a data base trigger?
a) ALTER TRIGGER
b) ALTER DATABASE
c) ALTER TABLE
d) MODIFY Trigger

Q.79 In SQL, which command(s) is (are) used to issue multiple CREATE TABLE, CREATE VIEW and GRANT statements in a single transaction?
a) CREATE PACKAGE
b) CREATE SCHEMA
c) CREATE. CLUSTER
d) All of the above

Q.80 In SQL, which command(s) is (are) used to remove rows from a table.
a) DELETE b) REMOVE
c) TRUNCATE d) Both a) & c)

Q.81 In SQL, which of the following is not a data definition language commands
a) RENAME b) REVOKE
c) GRANT d) UPDATE

Q.82 For a data base relation R (a, b, c, d), where the domains of a, b, c, d. include only atomic values, only the following functional dependencies and those that can be inferred from them hold a→c b→d. The relation is
a) the first normal form but not in second normal form
b) in second normal form but not in third normal form
c) in third normal form
d) none of the above.

Q.83 If a piece of data is stored in two places in the data base, then
a) storage space is wasted
b) changing the data is one spot will cause data inconsistency
c) it can be more easily accessed
d) both a)and b)

Q.84 A compound key
a) is made up a several pieces of information
b) uniquely identifies an item in a list
c) both a)and b)
d) is a combination of each unique key

Q.85 A data dictionary doesn't provide information about
a) where data is located
b) the size of the disk storage device
c) who owns or is responsible for the data
d) how the data is used

Q.86 What name is given to the collection of facts, items of information or data' which are related in some way?
a) Data base
b) Directory information
c) Information tree
d) Information provides

Q.87 Which of the following free from data base of IBM and compatible personal computers?
a) memory Mate b) Q and A
c) Paradox . d) Double Helix

Q.88 A report generator is used to
a) update files
b) print files on paper
c) data entry
d) all of the above

Q.89 When you have finished entering information into a form?
a) the template is written to the data file
b) the contents of the form are written to the data file
c) the contents of the form can be printed
d) all of the above

Q.90 What is the language used by most of the DBMSs for helping their users to access data?
a) High level language
b) Query language
c) SQL
d) 4GL

Q.91 A number of related records that are treated as a unit is called
a) file b) field
c) data d)batch

Q.92 Updating a data base means
a) revising the file structure
b) re-organizing the data base
c) modifying or adding record occurrences
d) all of the above

Q.93 Information can be transferred between the DBMS and a
a) spread sheet program
b) word processor program
c) graphics program
d) all of the above

Q.94 If a calculation is embedded in a form
a) the result of the calculations are stored with the form
b) the calculations are stored with the form
c) the result of the calculations are printed in report
d) all of the above

Q.95 A large computer information system maintains many different computer files. Which amongst them is called a perpetual file?
a) Specialized file b) Log file
c) master file d) Update file

Q.96 When using a data base management system, the first thing that you must do is to
a) create a data base file
b) activate file editor
c) load the software into your micro- computer
d) keep a floppy disk in readiness

Q.97 Data encryption techniques are particularly useful for
a) reducing storage space requirements
b) improving data integrity
c) protecting data in data communication systems
d) all of the above

Q.98 Long-range planning report produced in an MIS are primarily designed for
a) top management
b) middle management
c) lower management
d) all of the above

Q.99 Queries to a database
a) are written in English
b) can use aggregate functions like SUM and COUNT
c) both a)and b)
d) all of the above

Q.100 Versatile report generators can provide
a) Columnar totals
b) Sub-totals
c) Calculations
d) All of the above

Q.101 The files stored on a secondary stage device are composed of a hierarchy of data. What does a record in a file contain?
a) Bits b) Characters
c) Data field d) Schema

Q.102 The language used in application programs to request data from the DBMS is referred to as the
a) DML
b) DDL
c) QUERY language
d) All of the above

Q.103 Which of the following is a type of DBMS software?
a) data base manipulation language
b) query language
c) utilities
d) report writer

Q.104 The data dictionary tells the DBMS
a) what files are in the data base
b) what attribute are possessed by the data
c) what these files contain
d) all of the above

Q.105 To have a file hold a list, it is necessary to
a) identify the records in the list
b) identify the name, width and type of the fields of each record
c) decide which fields will be used as sort or index keys
d) all of the above

Q.106 With respect to data input, the most accurate description of batch control is

a) dividing documents to be entered into batches
b) controlling the input of each input clerk
c) comparing to a pre-calculated figure the total of a data item summed across a batch records put into the system .
d) all of the above

Q.107 The relational model uses some unfamiliar terminology. A tuple is equivalent to a
a) record b) field
c) file d) data base

Q.108 Which of the following is not an advantage of the data-base approach?
a)elimination of the data redundancy
b) ability to associate related data
c) increased security
d) all of the above are advantages.

Q.109 Which of the following command is (are) used to recompile a stored procedure in SQL?
a) COMPILE PROCEDURE
b) ALTER PROCEDURE
c) MODIFY PROCEDURE
d) All of the above

Q.110 In SQL, which command(s) is (are) used to enable/disable all triggers on a table?
a) ALTER TRIGGERS
b) ALTER, TABLE
c) MODIFY TRIGGERS IN TABLE
d) All of the above

Q.111 Aggregation is
a) An abstraction through which relationships are treated as lower level entities.
b) An abstraction through which relationships are treated as higher level entities.
c) An abstraction through which relationships are not treated at all as entities.
d) All the above

Q.112 The collection of information stored in a database of a particular moment is.....
a) view b) schema
c) instance d) subschema

Q.113 Construct an E - R model for the below assumption and find out the degree of the relationship.
A technician uses exactly one notebook for each project. Each notebook belongs to one technician for each project. Note that a technician may still work on many project and maintain different notebooks for different projects.
a) unary b) binary
c) ternary d) two binary relations

Q.114 1:N relationships in E-R diagram is implemented in relational model as
a) foreign keys are added on both sides
b) relation corresponding to '1' is modified to include foreign key of the relation on the 'N' side
c) primary keys are added on bothsides
d) relation corresponding to 'N' side is modified to include foreign key of the relation on the '1' side

Q.115 An entity relationship diagram is a tool to represent
a) Data model
b) Process model
c) Event model
d) Conceptual model

Q.116 The process of designation sub grouping within an entity set is called
a) Specialization b) Generalization
c) Aggregation d) Instantiation

Q.117 The data in the database at particular moment of time is called
a) Schema b) Snapshot
c) Intension d) Extension

Q.118 A primary key for an entity is
a) A candidate key
b) Any attribute
c) A unique attribute
d) A super key

Q.119 Conceptual schema is transformed from high level data model into the implementation data model. This step is called
a) Data model mapping
b) Conceptual schema
c) Functional mapping
d) Conceptual operation

Q.120 Singleton is
a) A set with only one element
b) A set with a ton of element
c) A single value
d) A key attribute of an entity value

Q.121 Weak entity type is known as
a) Parent entity type
b) Dominant entity type
c) Child entity type
d) Co-ordinate entity type

Q.122 A super key for an entity consists of
a) One attribute only
b) At least two attributes
c) At most two attributes
d) One or more attributes

Q.123 Relation type are called
a) Association b) Aggregation
c) Links d) Discrimination

Q.124 Link attribute is
a) A relationship attribute
b) A discriminator
c) A relationship tuple
d) A link tuple

Q.125 Ontology is similar to
a) Physical schema
b)Logical schema
c)Conceptual schema
d)Sub schema

Q.126 The diagram which depicts the flow of task between various components of a system is
a) Class diagram
b) Use case diagram
c) Activity diagram
d) Implementation diagram

Q.127 The diagram which shows the system components of their inter - connections, both at the software component level and the hardware component level is
a) Class diagram
b) Use case diagram
c) Activity diagram
d) Implementation diagram

Q.128 Entities of specified type can be grouped into,
a) Database b) Entity set
c) Attribute d) Table

Linked Answer Questions for Q. No. Q.129 & Q.130

Q.129 The primary keys are required for mapping following ERD:

a) 1 b) 2
c) 3 d) 4

Q.130 How many foreign keys are required for above relations.
a) 1 b) 2
c) 3 d) 4

Q.131 Which multi valued dependency is not followed from the following table?

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

a) ENAME→ PNAME
b) ENAME→DNAME
c) PNAME →ENAME
d) None

Q.132 Consider the two relations student (name, rollno, deptno, rank) and

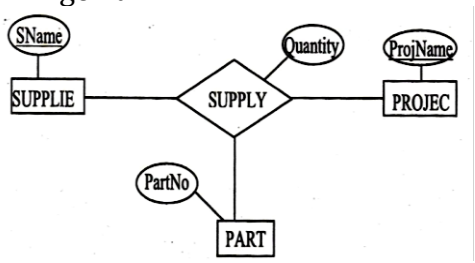
course (deptno, rollno, deptname, teacher, marks). Which of the following queries cannot be expressed using the basic relational algebra operations (U, -,x, π, σ, p)?

a) Rank of every student in a department
b) Students whose name is the same as their department name
c) The sum of all marks for a particular course
d) All students of a given department

Q.133 In one to many relationship, Entity type on one side of relationship called

a) Parent b) Child
c) Subtype d) Super type

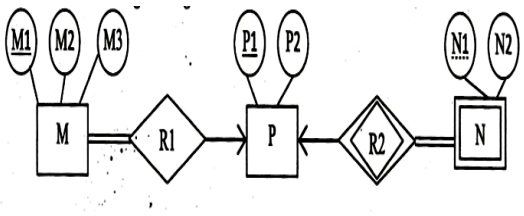
Q.134 What will be the number of relations if we follow ER-to-Relational Mapping Algorithm?



a) 3 b) 4
c) 5 d) 6

Statement for Linked Answer Questions Q.135 & Q.136

Consider the following ER Diagram



Q.135 The minimum number of tables needed to represent M, N, P, R1, R2 is

a) 2 b) 3
c) 4 d) 5

Q.136 Which of the following is a correct attribute set for one of the tables for the correct answer to the above question.

a) [M1, M2, M3, Pl.)
b) (M1, P1, N1, N2}
c) (M1, P1, N1)
d) (M1, P1)

Q.137 Indicate which of the following statements are true: A relation database which is in 3NF may still have undesirable data redundancy because there may exist:

a) transitive functional dependencies
b) non-trivial functional dependencies involving prime attributes on the right side
c) non-trivial functional dependencies involving prime attributes only on the left side
d) non-trivial functional dependencies involving only prime attributes

Q.138 Let us consider the following declaration R represents n entity and the set X, Y of attributes represents the key of R then if R represents a one-to-one relationship between entity E1 and E2 then which of the following is correct?

a) The FDY→ X will hold only
b) The FD X→Y will hold only
c) a) and b) hold together
d) Neither a) nor b) holds

Q.139 Given two sets of Function Dependencies (FDs) F and G over a relation scheme R if G covers F and if no proper subset G'(G'⊑G) covers F, then G is called

a) determinant cover
b) nonredundant cover
c) nondeterminant cover
d) redundant cover

Q.140 Spurious tuples are

a) the multiple entries with same value
b) the tuples with null values that exist in the original relation

c) the tuples with null values that did not exist in the original relations
d) the tuples with null values

Q.141 If F is a set of FDs on a relation scheme R then F+, the closure of F is the smallest set of FDs such that F+ is in 3NF and no FD can be derived from F by using the interference axioms that are not contained in F+
a) it is assumed to contain all the attributes that appear in F
b) whole statement is incorrect
c) F+ i.e. closure is not possible
d) none of these

Q.142 A primary key if combined with a foreign key creates z
a) many- to -many relationship between the tables that connect them
b) network model between the tables that connect them
c) parent child relationship between the tables that connect them
d) none of the above

Q.143 If all attributes are candidates key in relation, then it is in
a) 1 NF b) 2 NF
c) 3 NF d) BCNF

Linked Answer Questions for Q.No. Q.144 & Q.145
Suppose you are given relation R(X,Y,Z,W) with functional dependencies XY->Z, ZW->X and Y->W

Q.144 How many keys (minimal) are there in R?
a) 1 b) 2
c) 3 d) 4

Q.145 The highest normal forth of R is
a) 1NF b) 2NF
c) 3NF d) BCNF

Q.146 R(X, Y, Z, W) having FDs {X→Y; Y→Z; Z→W; W→X} So number of candidate Keys
a) 1 b) 2
c) 3 d) 4

Q.147 Second NF related with
a) Atomic Data
b) Full functional dependency
c) Option A and B are compulsory
d) None of these

Q.148 Which of the following can be used to represent "No seat is booked by more than one person"?
a) ∃s ∃p ∃q {[B(p,s)∧B(q,s)]->(p=q)}
b)∀s ∀p ∀q {[B(p,s)∧B(q,s)]->(p=q)}
c) ∃s ∃p ∃q{[B(p,s)UB(q,s)]->(p=q)}
d)∀s∀p ∀q{[B(p,s) U B(q,s)]->(p=q)}

Q.149 Suppose you are given a relation R with four attributes ABCD. For the set of FDs, assuming those are the only dependencies that hold for R, do the following:
C→D, C →A, B →C
Suppose, the relation is decomposed into 3 relations AC, BC, CD. Which of the following is true?
a) Lossless, Dependency preserving
b) Lossless, Not dependency preserving
c) Not lossless, Dependency preserving
d) Not lossless, Not dependency preserving

Q.150 For a database relation. R(a, c, d), Where the domains of a, b, c, d include only atomic values, only the following functional dependencies and those that can inferred from them hold:
a →c
b →d
The relation is
a) In first normal form but not in second normal form
b) In second normal form but not in third normal form
c) In third normal form
d) None of the above

Q.151 Let R(a, b, c) and S(d, e, f) be two relations in which d is the foreign key of S that refers to the primary key of R. Consider the following four operations on R and S

a) Insert into R b) Insert into S
c) Delete from R d)Delete from S

Which of the following statements is true about the referential integrity constraint above?

a) None of a), b), c) or d) can cause its violation
b) All of a), b), c) or d) can cause its violation
c) Both a) and d) can cause its violation
d) Both b) and c) can cause its

Q.152 Relational algebra is a _____ query language

a) Procedural b) Structural
c) Objected oriented d) None

Q.153 Let R1 (A, B, C, D) and R2 (D, E) be two relation schema, where the primary key sare, shown underlined, and let C be a foreign key in R1 referring to R2. Suppose there is no violation of the above referential integrity constraint in the corresponding relation instances r1 and r2 which one of the following relational algebra expressions would necessarily produce an empty relation?

a) $\prod_D(r2)-\prod_C(r1)$
b) $\prod_C(r1)-\prod_D(r2)$
c) $\prod_D(r1 \bowtie C \neq Dr2)$
d) $\prod_C(rl \bowtie C=Dr2)$

Q.154 Deposit (acno, cname, bal, bname)
Borrow (lno, cname, amt, bname)
Query 1:
{t/3S∈ Deposit (t [cname] = s [cname]) ∧∃u∈ Borrow (t [cname] = u [cname])}
Query 2:
(π_{cname} (Deposit ⋈ Borrow)
Query 3:
(π_{cname}(Deposit) π_{cname} (Borrow))
Query 4:
Select cname from Deposit d where exists (select * from Borrow .b where b .cname=d.cname).
Which of the following statements is/are false?

a) Query 2 is a natural join operation
b) Query 1 and query 2 gives same result
c) Both A and B
d) None

Q.155 Tuple relational calculus is a

a) Procedural language
b) Non - procedural language
c) Materialized language
d) Non - materialized language.

Q.156 Domain relational calculus is

a) Procedural language
b) Non - procedural language
c) Materialized language
d) Non - Materialized language

Q.157 Consider the following statements:
S1: ($\prod_{ename}$ (emp))
S2: Select ename from emp
Which of the following is true?

a) Both S1, S2 give same output
b) Duplicates may get printed in S1
c) Duplicates may get printed in S2
d) None

Q.158 In tuple relational calculus ∀t ∈r (A1(t)) equivalent to

a) →∃t ∈ r (→A1 (t))
b) Et (--,A1 (r))
c) E t r (-1A1 (t))
d)atEr(-1A1 (r))

Q.159 Consider the relation schemas as follows.
works(person name, company name, salary);
lives (person name, street, city);
located in (company name, city);
Relational algebra expression for

Find the names of the persons who work for company 'FBC' (company name='FBC').

a) $\pi_{\text{person name}}(\sigma_{\text{company name= FBC}}(\text{works}))$
b) $\pi_{\text{company name=FBC}}(\sigma_{\text{person name}}(\text{works}))$
c) $\pi_{\text{person name}}(\sigma_{\text{works}}(\text{company name} = \text{FBC}))$
d) $\pi_{\text{person name}}(\sigma_{\text{company name=FBC}}(\text{works X lives}))$

Q.160 Select the tuples for all employees who either work in department 4 and make over $ 25000 per year or each in department 5 and make over $ 30,000, refers to

a) σ(DNO=4 AND SALARY > 25000) OR (DNO = 5 AND SALARY > 30000)(EMPLOYEE)
b) π(DNO=4 AND SALARY>= 25000)OR (DNO = 5 AND SALARY > 30000) (EMPLOYEE)
c) σ(DNO=4 OR SALARY > 2500)OR (DNO=5 OR SALARY>3000) (EMPLOYEE)
d) σ(DNO=4 OR SALARY> 30000) OR (DNO=4 AND SALARY> 25000) (EMPLOYEE)

Q.161 Consider the Query SELECT student name FROM student WHERE class _name=(SELECT class-name FROM students WHERE math_marks=100); What will be the output?

a) The list of names of student with 100 marks in mathematics
b) The names of all students of all classes in which at least one student has 100 marks in mathematics
c) The names of all students in all classes having 100 marks in mathematics
d) The names and class of all students whose marks in mathematics is 100

Common Data for Question 162 and 163:
Consider the following database schema

SUPPLIERS(sid:integer,sname:string, address:string)
PART(pid:integer, pname:string, color:string)
CATALOG(sid:integer, pid:integer,cose:real)

Q.162 What does the following SQL query return?

SELECT c.sid FROM CATLOG c, PARTp WHERE (p.color ='red' OR p.color = 'green') and p.pid = c.pid

a) Find sids of suppliers who supply some red and green part
b) Find the sids of suppliers who supply some red or green parts
c) Find sids of suppliers who supply every red and every green part
d) Find sids of suppliers who supply every red or every geeen part

Q.163 What does the following SQL query return?

SELECT s.sid FROM SUPPLIERS s where s.address = '221 packer street' OR s.sid IN (SELECT c.sid FROM PART p, CATALOG c where p.color='red' AND p.pid = c.pid)

a) Find sid of supplier who supply some red part or are at 221 packer street
b) Find sid of supplier who supply some red part and at 221 packer street
c) Find sid of supplier who supply all red part and at 221 packer street
d) None of these

Q.164 Consider the query SELECT student name FROM student data WHERE roll no in (SELECT roll no FROM student marks WHERE SEM1_MARK=SEM2 MARK); Which of the following is true?

a) It gives the name of the student whose marks in semester1 and semester2 are not same
b) If gives all the names and roll no's of those students whose

marks in semester 1 and semester 2 are same

c) It gives the names of all the students whose marks in semester 1 and semester2 are same

d) It gives roll no's of all students whose marks in semester1 and semester2 are same.

Q.165 Consider the employee table EMP (id, salary) with id as primary key. Consider the following queries for obtained an id, with k^{th} highest sal (with a given k)

Query 1.
```
Select id
from EMP E1
where k = (select count (*) from EMP E2 where E2. Ei.sal)
```

Query2
```
Select id
from EMP E2
where k -1 = (select count (*) from EMP E2 where E2. SAL> Ei.sal)
```

a) Both query 1 and query2 are correct

b) Only quer1 is correct

c) Only query2 is correct

d) None of the above

Q.166 A time stamp is a combination of

a) Data and time

b) Data and day

c) Month and year

d) Second and fraction of second

Q.167 Consider the set of relations shown below and the SQL query that follows. Students : (Roll - number, Name, Data - of - birth)
Courses : (Course number, course - name, instructor)
Grades : (Roll - number, Course - number, Grade)
Select distinct Name from Students, Courses, Grades where students. Roll - number = Grades .Roll - number and courses . Instructor = Korth and Courses . Course - number = Grades . Course - number and Grades . grade = A
Which of the following sets is computed by the above query?

a) Names of students who have got an A grade in all courses taught by Korth

b) Names of students who have got an A grade in all courses.

c) Names of students who have got an A in at least one of the courses taught by Korth

d) None of the above

Q.168 SQL query is SELECT A FROM B WHERE C can be expressed as.

a) $\pi_C (\sigma_A b))$ b) $\pi_A (\sigma_C b))$

c) $\pi_B (\sigma_A c))$ d) $\pi_C (\sigma_B a))$

Q.169 Consider the following schedule S
S : R1a) R2b) Commit 2 R3c) W1a) R3a) W1b) Rollback 1 W3a) R4b)
What is the category of the schedule with respect to recoverability?

a) Recoverable schedule

b) Cascading rollback schedule

c) Non-recoverable schedule

d) Cascadeless rollback schedule

Q.170 Consider the following relational schema:
Suppliers(sid:integer, sname:string, city:string, street:string)
Parts(pid:integer, pname:string, color:string)
Catalog(sid:integer, pid:integer, cost:real)
Consider the following relational query on the above database:

```
SELECT S.srame
FROM Suppliers S
WHERE S.sid NOT IN (SELECT C.sid
FROM Catalog C
WHERE C.pid NOT (SELECT P.pid
FROM Parts P WHERE P.color< > 'blue'))
```

Assume that relations corresponding to the above schema are not empty. Which one of the following is the correct interpretation of the above query?

a) Find the names of all suppliers who have supplied a blue part
b) Find the names of all suppliers who have not supplied a blue part
c) Find the names of all suppliers who have supplied a non-blue part.
d) Find the names of all suppliers who have not supplied a non-blue part.

Q.171 Following statements belongs to DDL

a) Create b) Drop
c) Alter d) All of above

Q.172 Which of the following is valid SQL for an Index?

a) CREATE INDEX ID;
b) CHANGE INDEX ID;
c) ADD INDEX ID;
d) REMOVE INDEX ID;

Q.173 The HAVING clause does which of the following?

a) Acts like a WHERE clause but is used for groups rather than rows.
b) Acts like a WHERE clause but is used for rows rather than columns.
c) Acts like a WHERE clause but is used for columns rather than groups.
d) Acts EXACTLY like a WHERE clause.

Q.174 Consider two large files for keeping STUDENT and DEPT records, maintained in a database. Each STUDENT and DEPT records are of size 100 bytes and 20 bytes respectively. The STUDENT and DEPT files contain 30000 and 200 records respectively. Assume Disk blocks size of 1024 bytes and main memory buffers size is 7 blocks. Then which of the following query plan is cost effective (considering nested loop join strategy)?

a) STUDENT $|X|_{S.DEPTNO=D)DEPTNO}$ DEPT
b)DEPT$|X|_{s.DEPTNO=D)DEPTNO}$ STUDENT
c) $\sigma_{S.DEPTNO=D)DEPTNO}$(STUDENT x DEPT)
d) ALL are equivalent cost effective

Q.175 In a database, a transaction P holds a shared lock S. If transaction Q also request for a shared lock on S then.

a) The system is in deadlock
b) It is granted to Q only if P release it
c) Immediately it will be granted
d) Immediately it will be rejected

Q.176 Consider a non serial schedule S has 4 transaction and the execution of operation is given below:
S: R1a), R4b), W3a), W2b), W4c), commit4, W3c), W1a) comit1, R2d), W3d), Rollback3;
The above given Schedule is _____.

a) Non recoverable schedule
b) Recoverable schedule
c) Cascading rollback schedule
d) Cascade less rollback schedule

Q.177 Consider the following non serial schedule:
S: R1a), R2b), R3c), W1a), R2a), W2b), R3b), W2a),R1c), W3b), commit1, commit2, commit3. It is not allowed under which of the following protocols?

a) Mutliversion Timestamp protocol
b) Basic Timestamp protocol
c) Basic 2PL
d) Rigorous 2PL

Q.178 The following property makes sure that concurrent transaction do not interact

a) Atomicity b) Consistency
c) Isolation d) Durability

Q.179 Atomicity is managed by
a) Transaction management component
b) Recovery management component
c) Concurrency control component
d) None

Q.180 Isolation is managed by
a) Transaction management component
b) Recovery management component
c) Concurrency control component
d) None

Q.181 The timestamp schedule which accepts view serializable schedule is:
a) Basic timestamp
b) Thomas write rule
c) Both a) and b)
d) None

Q.182 In a multiple granularity protocol, et us assume that a Transaction T1 holds a data item 'D' in Intensive exclusive mode. Then which of the following lock requested by another transaction T2 can't be granted.
a) Intension shared
b) Intension exclusive
c) Both A and B
d) Shared

Q.183 Consider the relation R(W X Y Z) with the FD set F={X→Z,Z→X,WX→Y}.
Which of the following is true?
a) {R1(XZ),R2(WXY)} forms BCNF decomposition for R and it satisfies both lossless and dependency preserving conditions
b) {R1(XZ),R2(WYZ)} forms BCNF decomposition for R and it will satisfy lossless join condition but dependencies are not preserved.
c) Neither A nor B
d) Both A and B

Q.184 In conservative 2 phase locking protocol,
a) All exclusive locks are released before transaction commits
b) All locks must held before transaction
c) Create read & write list before holding any lock
d) None of these

Q.185 Consider the following transaction

T_1	T_1
Read items (X); X = X-N; Write item (X); Read items (Y); Y/: = Y+N; Write item (Y);	 Read item (X); X = X+ N; Write item (X);

Suppose the number of seats in a flight is 80.If there were so reservation on the flight (X), transfers 5 seat reservations for the flight corresponding to X to the flight corresponding to Y and reserves 4 seats on X. then what is final result should come and what has come according to the transaction?
a) 70, 84 b) 79, 84
c) 84, 79 d) None of these

Q.186 Which of the following statements is false for Time-stamp ordering protocol of database concurrency control management?
a) It always ensures conflict serializability property
b) It always ensures deadlock free

c) It always ensures recoverable schedule
d) Both B and C

Q.187 Consider three transactions Ti, T2 and T3, executing in a RDBMS environment according to the following schedule S:<R1a), W1a), R3(13), R3c), WO), W3c), committ(T3), R2b), R2a), W2a), W2b), committ(T2), R1c), W1c), committ(T1)>. The schedule S is
a) Conflict- serializable to <T1,T3,T2> but not recoverable
b) Conflict-serializable to <T3,T1,T2> and recoverable
c) Conflict-serializable to <T1,T3,T2> and recoverable
d) Conflict-serializable to <T3,T1,T2> but not recoverable

Q.188 The solution to the recurrence relation $T(n) = 4T\left(\left\lfloor \sqrt{n} \right\rfloor\right) + \log n$ is
a) O(lognloglogn) b) O(loglogn)
c) O(log2n) d) O(logn2)

Q.189 An indexing operation is
a) Sorting a file using a single key
b) Sorting a file using two keys
c) Ordering of the record based on search key value
d) Both (b) and (c)

Q.190 Index consists of
a) A list of keys
b) Pointer to the record
c) Both b) and a)
d) None

Q.191 Main advantage of B tree is
a) reduce searching time
b) reduce redundant storage of search key value.
c) Increase through put of the system
d) None

Q.192 In sequential file organization if a record is not fit in free space then the record is placed in
a) Last block b) First block
c) Overflow block d) None

Q.193 A B+ tree of order n index file each leaf node must contain
a) n -1 values b) n - 2 values
c) n/2 values d)(n-1)/2 values

Q.194 Smallest indivisible unit of magnetic disc is called as
a) Cylinder b) Track
c) Platter d) Sector

Q.195 Each block of B-Tree holds space for,
a) n search key, n+1 pointer
b) n + 1 pointer, n search key
c) n search keys n pointer
d) cannot be determined

Common Data for Q. No. 196 & 197
Consider a DBMS that has the following characteristics:
2KB fixed-size blocks
12-byte pointers
56-byte block headers
We want to build an index on a search key that is 8 byte long.

Q.196 What is maximum number of records we can index with a 3-level (2 levels plus root) B+ Tree.
a) 990000 b) 100000
c) 9900 d) None of the above

Q.197 What is maximum number of records we can index with a 3 level B Tree
a) 238327 b) 384399
c) Both A or B d) None of the above

Q.198 Which one of the following statements about normal forms is FALSE?
a) BCNF is stricter than 3NF
b) Lossless, dependency-preserving decomposition into 3NF is always possible
c) Lossless, dependency-preserving decomposition into BCNF is always possible
d) Any relation with two attributes is in BCNF

Common Data for Q. No. Q.199 & 200

Suppose the following two processes, foo and bar are executed concurrently and share the semaphore variables S and R (each initialized to 1) and the integer variable x (initialized to 0).

```
void foo ( ){
    do {
        semWait (S);
        semWait(R);
        x++;
     semSignal (S);

     semSignal(R);
      } while (1);
 }
```

```
void bar ( ){
    do {
        semWait (S);
        semWait(R);
        x--;
        semSignal (S);
        semSignal(R);
        } while (1);
}
```

Q.199 Choose the correct option

a) Concurrent execution of these two processes can result in one or both being blocked forever
b) It cannot lead to deadlock
c) Incorrect use of semWait()
d) Incorrect use of semSignal() .

Q.200 Choose the correct option

a) Concurrent execution of these two postponement of one of them
b) Concurrent execution of these two postponement of one of them
c) Incorrect use of Semwait()
d) Incorrect use of Semsignal()

ANSWER KEY:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(b)	(a)	(a)	(a)	(c)	(b)	(a)	(a)	(c)	(b)	(b)	(a)	(b)	(c)	(a)
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
(c)	(a)	(a)	(c)	(c)	(c)	(c)	(d)	(c)	(c)	(d)	(c)	(a)	(a)	(a)
31	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**
(a)	(a)	(a)	(a)	(a)	(a)	(c)	(a)	(d)	(b)	(b)	(b)	(a)	(d)	(a)
46	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
(a)	(b)	(a)	(d)	(b)	(b)	(d)	(d)	(b)	(c)	(a)	(b)	(b)	(d)	(d)
61	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**	**75**
(a)	(a)	(a)	(b)	(a)	(c)	(c)	(b)	(d)	(d)	(d)	(c)	(c)	(a)	(a)
76	**77**	**78**	**79**	**80**	**81**	**82**	**83**	**84**	**85**	**86**	**87**	**88**	**89**	**90**
(d)	(d)	(c)	(b)	(d)	(d)	(a)	(d)	(c)	(b)	(a)	(a)	(b)	(b)	(b)
91	**92**	**93**	**94**	**95**	**96**	**97**	**98**	**99**	**100**	**101**	**102**	**103**	**104**	**105**
(a)	(c)	(d)	(b)	(c)	(c)	(c)	(a)	(b)	(d)	(c)	(a)	(a)	(d)	(d)
106	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**	**120**
(c)	(a)	(d)	(b)	(b)	(b)	(c)	(c)	(d)	(d)	(a)	(b)	(c)	(a)	(a)
121	**122**	**123**	**124**	**125**	**126**	**127**	**128**	**129**	**130**	**131**	**132**	**133**	**134**	**135**
(c)	(d)	(c)	(a)	(c)	(c)	(d)	(b)	(b)	(b)	(c)	(c)	(a)	(b)	(b)
136	**137**	**138**	**139**	**140**	**141**	**142**	**143**	**144**	**145**	**146**	**147**	**148**	**149**	**150**
(a)	(a)	(c)	(b)	(c)	(a)	(c)	(d)	(b)	(a)	(d)	(c)	(b)	(a)	(a)
151	**152**	**153**	**154**	**155**	**156**	**157**	**158**	**159**	**160**	**161**	**162**	**163**	**164**	**165**
(d)	(a)	(b)	(b)	(b)	(b)	(c)	(a)	(a)	(a)	(a)	(b)	(a)	(a)	(b)
166	**167**	**168**	**169**	**170**	**171**	**172**	**173**	**174**	**175**	**176**	**177**	**178**	**179**	**180**
(a)	(c)	(c)	(b)	(b)	(d)	(a)	(a)	(b)	(c)	(d)	(d)	(c)	(a)	(c)
181	**182**	**183**	**184**	**185**	**186**	**187**	**188**	**189**	**190**	**191**	**192**	**193**	**194**	**195**
(c)	(d)	(d)	(b)	(c)	(d)	(d)	(a)	(c)	(c)	(b)	(c)	(d)	(d)	(a)
196	**197**	**198**	**199**	**200**										
(a)	(c)	(c)	(a)	(b)										

EXPLANATIONS

Q.1 **(b)**
Adequate level of database design is 3NF.

Q.4 **(a)**
If Y ⊆ X, then X→Y is trivial.

Q.5 **(c)**
If all non-key attributes depends on primary key, it is 3NF definition.

Q.7 **(a)**
First selection then projection must be applied.

Q.11 **(b)**
a, c can be get with transitive property.

Q.14 **(c)**
The join returns a record even if we have one matching.

Q.16 **(c)**
The same flight number flies on different dates. So the combination is key.

Q.17 **(a)**
ab combination is the key. Since c is depending on part of the key, it is not in 2NF.

Q.18 **(a)**
Maximum and Minimum would be only the students=(8,8) Since student table is parent and enroll is child.

Q.21 **(c)**
When parent is deleted, then child is deleted. The converse is not true.

Q.23 **(d)**
Subquery returns salary of the company and main query gets average salary of male employees department wise.

Q.24 **(c)**
Relational Algebra is procedural language, Relational Calculus is non-procedural language.

Q.25 **(c)**
Hierarchical model is based on graphs and Network model is based on trees.

Q.26 **(d)**
One student can enroll for more than one course. And one course is opted by more than one student.

Q.28 **(a)**
Alternate key is secondary key. It is a candidate key other than primary key.

Q.31 **(a)**
The problems due to redundancy are consistency and integrity.

Q.33 **(a)**
The primary key of one table becomes foreign key in another table.

Q.36 **(a)**
Check constraint is integrity constraint.

Q.39 **(d)**
Trigger automatically executes when condition associated with it matches when DB modification is done.

Q.42 **(b)**
Emp_no is primary key which can identify project_no.

Q.43 **(a)**
Since emp-name behaves like a key, all attributes are dependent on it.

Q.44 (d)
Insertion into child table and deletion from parent table can cause violation.

Q.46 (a)
Selection operation is commutative.

Q.49 (d)

Given FD's	Step:1	Step:2	Step:3
A→BC	A→B, A→C	A→B, A→C	A→B
B→C	B→C	B→C	B→C
A→B	A→B	A→B	Remove A→B, since repeated twice.
AB→C	AB→C	A→C	A→C can be removed, Since it can be derived From A→B and B→C by Transitive Rule.

Q.50 (b)
Shared locks can be shared by different transactions simultaneously.

Q.112 (c)
Information at a particular moment in a data base is called instance.

Q.113 (c)
Degree of relationship is ternary because relationship will keep three entity technicians, notebook and project.

Q.114 (d)
Relation corresponding to 'N' side is modified to include foreign key of the relation on the '1' side

Q.115 (d)
E-R model take place in conceptual database model.

Q.116 (a)
Sub grouping is called specialization

Q.117 (b)
In the database, a snapshot is the state of a database at a particular point of time.

Q.118 (c)
Primary Key should be a unique attribute.

Q.119 (a)
This step is called Data model mapping

Q.120 (a)
A set with only one element

Q.121 (c)
It is known as child entity type

Q.122 (d)
Super key can consist one or more attribute.

Q.123 (c)
Relationship instances are called links.

Q.124 (a)
Link attribute is a relationship attribute

Q.125 (c)
It is similar to Conceptual schema. An ontology is a specification of a conceptualization.

Q.126 (c)
It is called activity diagram. Activity diagrams are graphical representation of workflows of stepwise activities and action.

Q.127 (d)
Implementation diagram shows the system components inter - connection.

Q.128 (b)
Set of similar entity set called entity set.

Q.129 (b)

Primary keys are present in student table of book table only

Q.130 (b)

Only 2 foreign keys placed in issues table

Q.131 (c)

A multi valued dependency (MVD) X -- > > Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation state r of R: If two tuples t1 and t2 exist in r such that t1[X]=t2[X], then two tuples t3 and t4 should also exist in r with the following properties, where we use Z to denote (R2 (X u Y)):
t3 [X] = t4 [X] and t1 [X] = t2 [X]
t3 [Y] = t1[Y] and t4 [Y] = t2 [Y]
t3 [Z] = t2 [Z] and t4 [Z] = t1[Z]

Q.132 (c)

The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product(x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations.

Q.133 (a)

In a one-to-many relationship, the entity that is on the one side of the relationship is called a parent entity.

Q.134 (b)

SUPPLIERR

SNAME	...

PROJECT

PRONAME	...

PART

PARTNO	...

SUPPLY

SNAME	PRONAME	PARTNO	QUANTITY

Q.135 (b)

Three tables will be formed
M(M_1, M_2, M_3, P_1)
P (Pi, P2)
N (N_1, N_2, P_1)

Q.137 (a)

Transitive functional dependencies.
If all the transitive dependencies are not eliminated, we may have to use null value to represent some of the possible meaningful relationships among data items, and there is the problem of repetition of information.

Q.138 (c)

For the given situation in the question, FD X→Y and Y→X will holds together since R represents a one to one

Q.140 (c)

Spurious tuples are the tuples with null values that did not exist in the original relations.

Q.141 (a)

The explanation given in the if R is not specified then it is assumed to contain all the attributes that appear in F.

Q.143 (d)

BCNF is when all determinants are candidates key.

Q.144 (b)

XY and YZ

Q.145 (a)

Here X, Y, Z are prime attributes. Non prime attribute W depends on Y, which is not a candidate key. So, Y->W violates the definition of 2NF. Hence, the highest normal form is 1NF

Q.146 (d).
Candidates keys are X, Y, Z, W

Q.147 (c)
NF requires data in 1NF and full functional dependent.

Q.149 (a)
a) Candidate keys: B
b) C→D and C→A both cause violations of BCNF. One way to obtain a (lossless) join preserving decomposition is to decompose R into AC, BC and CD.

Q.150 (a)
a→ c
b →d holds so it is in INF
∴ It cannot be in 2 NF
All the values must be atomic values that means a prime 1- value must point to only one value not a list of values. Hence, a, b, c, d include only atomic values only the following functioned dependencies.

Q.151 (d)
When insert into 'S' operation will take place there will be inconsistency in the database, since it has a foreign key which refers to the primary key of R. When delete from 'R' take place it will cause violation because since it's primary key is the foreign key for 'S' so there will be inconsistency in the database. Before deleting from R, the corresponding rows from S has to be deleted. The other option a) and d) will not cause such problems.

Q.152 (a)
It is procedural language.

Q.154 (b)
Query 1 and 2 will give same result

Q.155 (b)
It is a non - procedural language

Q.156 (b)
Domain relational calculus non-procedural language.

Q.157 (c)
Select statement retains duplicates where as 11 gives distinct output.

Q.159 (a)
$\pi_{\text{person name}}(\sigma_{\text{company name=FBC}}(\text{works}))$
SQL: Select person_name
From works
Where company_name = 'FBC'

Q.162 (b)
Given SQL query find the sids of suppliers who supply some red or green parts.

Q.163 (a)
Sid of supplier who supply some red part or at 221 packer street.

Q.165 (b)
For the salary field of each tuple in the outer Queries of Quer 1 & Query2, The Sub query in Query 1 & Query2 outputs the number of salaries that are less than or equal to the salary field of the tuple of the outer query By which query 1 gives us the correct output.

Q.166 (a)
It is a combination of date and time.

Q.169 (b)
In transaction 1, A and B variables modified and after that modification transaction 3 has read it so if we rollback transaction 1 we should

rollback transaction 3 also. So, it is cascading rollback schedule.

Q.172 (a)

CREATE INDEX ID is valid SQL for an Index

Q.173 (a)

HAVING clause acts like a WHERE clause but is used for groups rather than rows.

Q.174 (b)

Total no of block access =
$B_D + [B_D/(n_s - 2)] * B_D$ [in case B]

Q.175 (c)

The new shared lock will be immediately granted because the previous lock was also shared lock

Q.176 (d)

The given schedule is cascade less rollback schedule.

Q.177 (d)

It is not allowed under rigorous 2PL

Q.178 (c)

Isolation makes sure concurrent transaction do not interact.

Q.179 (a)

Atomicity is managed by transaction management system.

Q.180 (c)

Isolation is managed by concurrency control component.

Q.183 (d)

keys = {(WX),(WZ)}⇒{WX, WZ}
X→Z,Z→Xviolates BCNF
so {(XZ), (WXY)} and {(XZ), (WYZ)} BCNF decomposition
so both option A & B are correct after checking the lossless and dependency preserving conditions.

Q.184 (b)

In Conservative 2phase locking, transactions obtain all the locks they need before the transactions begin.

Q.185 (b)

If X=80 at the start (originally there were 80 reservations on the flight), N=5 (T1 transfers 5 seat reservations from the flight corresponding to X to the flight corresponding to Y), and M=4 (T2 reserves 4 seats on X), the final result should be X=79; but it is X=84 because the update in T1 that removed the five seats from X was lost. It is a lost update problem.

Q.189 (c)

Indexing→ordering of the record based on search key value.

Q.190 (c)

Index→list of key, pointer to the record

Q.191 (b)

Its main advantage is to reduce the redundant storage of search key value.

Q.192 (c)

In overflow block

Q.193 (d)

$\frac{n-1}{2}$ values

Q.194 (d)

Q.195 (a)

n-Pointer with B-Tree index & each block have space for n- search key values & n + 1 pointer

Q.196 (a)

Let each node of a B+ tree contains at most n pointers and hence (n-1) keys So $8*(n-1)+12*n+56 \le 2048$
so $n \le 100$
so maximum leaf level pointers (record)=99*100*100

so maximum number of records that can be indexed 990000

Q.197 (c)

$8 * (n-1)+12 * (2n-1)+56 \leq 2048$
so $n \leq 62$
First level of B- tree can hold 61 records pointers second level can hold at maximum 62 * 61 record pointers so leaf node have 62 * 62 * 61 record pointers, maximum number of records (indexed)= 61 + 62 * 61+62*62*61
= 238327
But if we assume that each leaf node of B- tree hold at most 99 record pointers then Maximum number of record (indexed) = 384399

Q.198 (c)

Lossless, dependency preserving decomposition into BCNF is not always possible

Q.199 (a)

If foo() executes sem Wait(S) and then bar() executes sem Wait(R) both processes will then block when each executes its next instruction. Since each will then be waiting for a semSignal() call from the other, neither will ever resume execution.

Q.200 (b)

If either process blocks on a semWait() call then either the other process will also block as described in (a) or the other process is executing in its critical section. In the latter case, when the running process leaves its critical section, it will execute a semSignal()call, which will awaken the blocked process.