DBMS – Assignment 1

V. Manvitha
19131A0596
CSE-4

(1)

(a) Explain the following with example

(i) Degree (ii) Participation (iii) Cardinality

For ER model

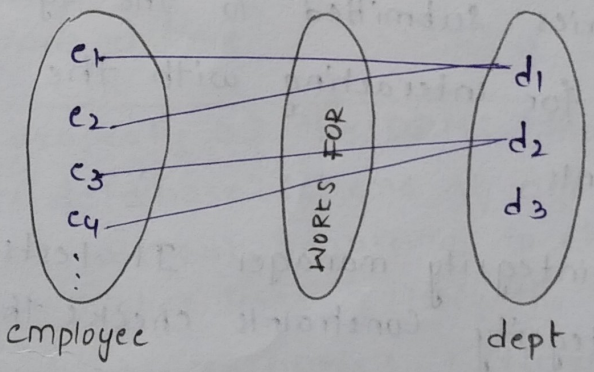
Degree: It is the number of entity types that participate (associate) in a relation ship

Ex:- A relationship is defined as

Every employee works for exactly one department

A department can have many employees

A new department need not to have any employee



In the above example, there are 2 entity types Employee, dept so <u>degree = 2</u>

Cardinality: Cardinality of a relationship is the maximum no. of relationships in which an entity can participate

Ex:- In the above example

<u>Cardinality of employee = 1</u> (An employee can work for only one dept)

<u>Cardinality of department = Many</u> (A department can have many employees)

participation: it is the minimum number of relationships in which an entity can participate

Ex:- In the above example

participation of employee = 1 (An employee belongs to a department)

participation of department = 0 (A new dept need not to have any employee)

(b) Explain storage manager and Query processor.

These are the Components of DBMS

Storage manager: It provides the data blw the low level data stored in the database and application program and queries submitted to the system.

It is responsible for interacting with the file system

It has 4 Components

Authorization and integrity manager: It tests for the satisfaction of integrity constraints checks the authority of users to access data.

Transaction manager: It ensures that the DB remains in a consistent state despite the system failures and that Concurrent transaction execution proceeds with conflicting

File manager: It manages the allocation of space on disk storage and the data structures used to represent information stored on a disk.

Buffer manger: It is responsible for fetching data from disk storage into main memory and deciding what data

to cache in memory.

Query processor: It has 3- Sub components

(i) DML Compiler :- It takes DML queries and prepares a evaluation plan to execute query

It chooses the lowest execution cost plan & handovers to query engine execution

(ii) Query engine execution :- It executes low-level instructions generated by the DML Compiler.

(iii) DML Interpetor :- It interprets the DDL statements & records them in a set of tables Containing meta data (or) data dictionary

(2) Data independence

It is the property of the DBMS that helps you to change the database schema at one level of a data base system without requiring to change the schema at the next higher level. Justify the importance of Data independence in DBMS? Differentiate the logical and physical data independence

Ans:- Importance of data independence in DBMS:

→ It helps to improve the quality of the data
→ Database System maintainance will become affordable
→ No need to alter data structure in application programs.
→ It allows to improve state which is Undamaged (or) undivided

→ Easily make modifications in the physical level is needed to improve the performance of the system

Logical data independence

Logical data independence refers to the characteristic of being able to change the Conceptual schema without having to change the external schema.

→ logical data independence is used to seperate the external lvel from the Conceptual view.

→ If we do any changes in the Conceptual view of data, then the user view of the data will not get effected

→ It occurs at the User interface level

physical data independence.

physical data independence can be defined as the capacity to change the internal schema without having to change the Conceptual Schema.

→ It seperates Conceptual levels from the internal levels.

→ If we do any Changes in the storage size of the database System Server, then the Conceptual structure of the database will not be affected.

→ It occurs at logical interface level

(2)

(b) Write the advantages of DBMS over the file system

It is Computerized System, whose overall purpose is to maintain the information and make that information

available on demand, where as file system is a collection of data. In this system, the user has to write the procedures for managing the database

Advantages:

1) Redundancy can be reduced
2) Inconsistency can be avoided
3) Data can be shared
4) Standards can be enforced
5) Security restrictions can be applied
6) Integrity can be maintained
7) Data gathering can be possible
8) Requirements can be balanced

3) a)

I find all the sailors name whose rating is greater than 4

TRC: ~~{T | T ∈ Sailors ∧ (∃~~

$\{T \mid \exists T_1 \in Sailors\ (S_1.rating > 4 \wedge T.Sname = T_1.Sname)\}$

Relational algebra:

$\Pi_{Sname}(\sigma_{rating>4}(sailors))$

SQL:

Select * from Sailors natural join reserves where bid=101.

II find all the Sailors who reserved boat no 101

SQL: Select Sname from Sailors where rating>4;

TRC: $\{T \mid T \in Sailors \wedge (\exists R \in Reserves(R.sid = T.sid \wedge R.bid=101))\}$

Relational Algebra: $\Pi(\sigma_{bid=101}(reserves \bowtie sailors))$

(b) Explain referential intigrity Constraints with example.

A referential intigrity Constraint is specified b/w two tables.

→ In Referential intigrity Constraints, if a foriegn key in Table1 refers to the primary key of Table2, then every value of the foriegn key in table1 must be null or be available then in table 2.

Example:-

Emp_name	Name	Age	D-no → foriegn key
1	Jack	20	11
2	harry	40	24
3	John	27	18
4	Devil	38	13

Not allowed as D-NO 18 is not defined as a primary key of table 2 & in table 1 D-no is a foriegn key defined

Relationships

Primary key

D-no	D-location
11	Mumbail
24	Delhi
13	Noida

(4)
(a) Explain Outer joins with examples

Outerjoin
- left outerjoin
- Right outer join
- full outer join

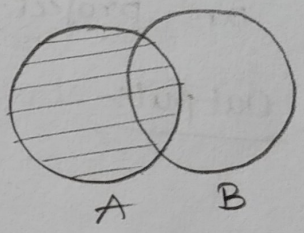
Left join: The SQL left join returns all the values from left table and the matching values from the right table if there is no matching join value, it will return NULL.

Ex:-

```
Select Employee.Emp_name, project.Department
    from Employee left join project
    on project.Emp_id = Employee.Emp_ID;
```

Sample output:-

Emp_name	Department
Angelina	Testing
Robert	development
christain	Designing
Russel	NULL
Marry	NULL

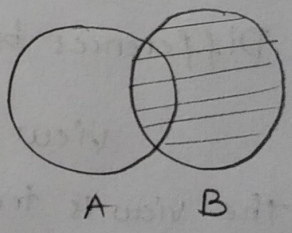


Right join: In SQL right join returns all the values from the values from rows of right table & the matched values from the left table. If there is no matching in both tables, it will return NULL.

Ex:-
```
Select Employee.Emp_name, project.Department
    from employee right join project
    on project.Emp_id = Employee.Emp_ID;
```

Sample output:-

Emp_name	Department
Angelina	Testing
Robert	Development
christain	Designing
NULL	Development

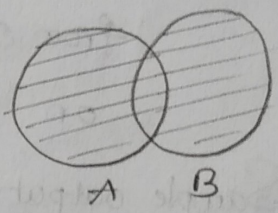


FULL JOIN

In SQL, full join is the result of combination of both left and right outer join. join tables have all the records from both tables. It puts NULL on the places of matches not found.

example:-

```
Select Employee. Emp-name, Project. department
from employee
full outer join project
on project. Emp-ID= Employee. Emp-ID;
```



Output:-

Angelina	Testing
Robert	Development
Christan	Designing
Russel	NULL
Marry	NULL
NULL	development

(b) What is the difference between View and Table? Write sql syntax to create a view.

Ans:- Syntax to create a view:

```
Create view view_name as
Select col1, col2--- from table_name
where condition;
```

Differences b/w View and table.

View	Table
(1) The view is treated as a virtual table that is extracted from a database	(1) The table is structured with a set no. of columns and a boundness of columns.

(2) A view is additionally a database object which is utilized as a table and inquiry that can be connected to different tables

(2) The table is databases which are utilized to hold the information that is utilized in application & reports

(3) The view is utilized to query certain information which is contained in a few distinct tables

(3) The table holds fundamental client information & holds cases of a characterized object

(4) Create view view-name as (select col1, col2--- from table_name where Condition);

(4) Syntax:- create table table_name (col1 datatype Constraints, col2 datatype Constraints, ---);

(5) (a) Closure of an Attribute:

(i) clousure of an attribute can be defined as a set of attributes that can be functionally determined from it.

clousure of a set F of FD's is the set F^+ of all FD's that can be inferred from F.

(i) Closure of $F = F^+ = \{F, I, J\}$ since $F \rightarrow IJ$

(ii) closure of $K = K^+ = \{K, M\}$ since $K \rightarrow M$

(iii) Closure of $L = L^+ = \{L, N\}$ Since $L \rightarrow N$

(ii) find all candidate keys

$EFH^+ = \{E, F, H, G, I, J, K, L, M, N\}$

So, all attributes are derived from EFH

So, EFH is a Candidate key.

(b) What is lossless decomposition? give example of lossy and lossless decomposition.

If the information is not lost from the relation that is decomposed, then the decomposition will be lossless.

The lossless decomposition guarentees that the join of relations will result in the same relation as it was decomposed

The relation is said to be lossless decomposition if natural joins of all the decomposition give the original relation.

Examples:-

lossless decomposition.

Consider a relation $R(A,B,C,D)$ is decomposed into

$R_1(A,B,C)$ and $R_2(C,D)$

Since R_1 and R_2 have a Common attribute C

when we do $R_1 \bowtie R_2$ it will give R

So it is a lossless decomposition.

lossy decomposition:

Consider a relation $R(A,B,C,D)$ is now decomposed into $R_1(A,B)$ and $R_2(C,D)$

Since R_1 and R_2 doesn't have a Common attribute Joins can't be performed, hence cartesian product Comes into action hence no.of tuples got increased and redundancy will occur.

So it is a lossy decomposition.

6. a) Explain the followings with example

I) Insertion Anomaly

II) deletion Anomaly.

Insertion Anomaly

Can't insert information whose tuple is inserted in referencing relation and referencing attribute value is not present in referenced attribute. it will not allow inserting in referencing relation.

Eg:- if a student detail has to be inserted whose Course is not being decided yet then insertion will not be possible till the time course is decided for student.

Student_id	Name	Contact	College	Course	Rank
100	Himanshu	7300934851	GEU		1

This problem happens when the insertion of data record is not possible without adding some additional unrelated data to the record.

Deletion Anamoly:

eg: If the details of students in this table is deleted then the details of college will also get deleted which should not occur by common sense.

This anamoly happens when deletion of a data record results in losing some unrelated information that was stored as a part of the record that was deleted from the table.

(b) partial functional dependency:-

partial functional dependency occurs when a non-prime attribute is functionally dependent on a part of a candidate key.

Mathematically, $X \to y$ is PD $\iff$ X is candidate key $\wedge$ y : NPA

Given relation is R (A, B, C, D, E, F)

ABC and ACD are candidate keys, since

$ABC^+ = \{A, B, C, D, E, F\}$ and $ACD^+ = \{A, B, C, D, E, F\}$

$\therefore$ A, B, C, D are prime attributes

E, F are non prime attributes

(i) Since $ABD \to E$

$ABD \subseteq ABD$ so it isn't a partial dependency.

(ii) F depends on DC

$DC \to F$

$CD \subset ADC$ (candidate key)

F depends of part of candidate key so its PFD

Hence the relation is not in 2NF

Here the partial dependency is $CD \to F$

so R_1 (A, B, C, D, E) and R_2 (C, D, F) decomposed

CD is a primary key in R_2 & foriegn key in R_1