CS-303 DBIS MIDTERM SOLUTIONS

1. PreparedStatement performance is better, PreparedStatement are precompiled and the query plan is created only once irrespective of how many times we are executing that query. PreparedStatement allows you to write a dynamic and parametric query and is faster than Statement in Java Statement preferred over PreparedStatement interfaces if we are executing a particular SQL query only once. PreparedStatement if we are executing a particular SQL query multiple times.
2. a.

   stmt = conn.createStatement()
   String query = "select company_name from Works group by company_name having avg(salary)>(select avg(salary) from Works where company_name='First Bank Corporation')";
   ResultSet rs = stmt.executeQuery(query);

   b.

   stmt = conn.createStatement()
   String query = "Update works set salary = (case
   When salary * 1.1 <100000 then salary*1.1
   Else salary * 1.03";
   ResultSet rs = stmt.executeQuery(query);

   c.

   stmt = conn.createStatement()
   String query = "Delete from works where company_name ='Small Bank Corporation'";
   ResultSet rs = stmt.executeQuery(query);

3. V

```
Select Stu_Id,Stu_Name,Marks,
Case When Marks > 32 Then 'Pass' Else 'Fail'
End as Remarks,
Case When Marks >= 76 Then 'AA'
     When Marks >= 65 Then 'AB'
     When Marks > 31 Then 'EB'
End as Grade
From Student
```

4.
   a. (σ fdate = 01/12/2022 ^ time = 16:00 (flight)) ∩ (σ fdate = 02/12/2022 ^ time = 16:00 (flight))
   b. (σ fdate = 09/11/2022 ^ time = 16:00 (flight)) ∪ (σ fdate = 22/11/2022 ^ time = 16:00 (flight))

c. Π aname (agency ⋈ (Π aid (agency) – Π aid (σ pid = 123 (booking)))
d. Π pname ((Π pid (passenger) - Π pid (booking)) ⋈ passenger)
e. Π passengers.pid, pname, pcity (σ pgender = “Male” (passengers ⋈ booking ⋈ agency))
f.

5.

A database is an integrated collection of data, usually so large that it has to be stored on secondary storage devices such as disks or tapes. This data can be maintained as a collection of operating system files, or stored in a DBMS (database management system). The advantages of using a DBMS are:

a)Data independence and efficient access. Database application programs are independent of the details of data representation and storage. The conceptual and external schemas provide independence from physical storage decisions and logical design decisions respectively. In addition, a DBMS provides efficient storage and retrieval mechanisms, including support for very large files, index structures and
query optimization.

b)Reduced application development time. Since the DBMS provides several important functions required by applications, such as concurrency control and crash recovery, high level query facilities, etc., only application-specific code needs to be written. Even this is facilitated by suites of application development tools available from vendors for many database management systems.

c)Data integrity and security. The view mechanism and the authorization facilities of a DBMS provide a powerful access control mechanism. Further, updates to the data that violate the semantics of the data can be detected and rejected by the DBMS if users specify the appropriate integrity constraints.

d)Data administration. By providing a common umbrella for a large collection of data that is shared by several users, a DBMS facilitates maintenance and data administration tasks. A good DBA can effectively shield end-users from the chores of fine-tuning the data representation, periodic back-ups etc.

e)Concurrent access and crash recovery. A DBMS supports the notion of a transaction, which is conceptually a single user’s sequential program. Users can write transactions as if their programs were running in isolation against the database. The DBMS executes the actions of transactions in an interleaved fashion to obtain good performance, but schedules them in such a way as to ensure that conflicting
operations are not permitted to proceed concurrently. Further, the DBMS maintains a continuous log of the changes to the data, and if there is a system crash, it can restore the database to a transaction-consistent state. That is, the actions of incomplete transactions are undone, so that the database state reflects only the actions of completed transactions. Thus, if each complete transaction, executing alone, maintains the consistency criteria, then the database state after recovery
from a crash is consistent.

If these advantages are not important for the application at hand, using a collection of

files may be a better solution because of the increased cost and overhead of purchasing and maintaining a DBMS.

6. One possible answer is A query Language (5) and Crash Recovery (3).

Feature no. 1 is not needed as only Gru will access the stand alone system.
Feature no. 2 is also not needed because concurrency control will be required when concurrent users try to access the database.
Feature no. 3 is required because large volumes of data will be stored in the database and if the system crashes then the crash recovery feature will recover the lost data.
Feature no. 4 is not required because only 1 user (Gru) will access the database
Feature no. 5 is required because data needs to be stored and retrieved with the help of query language.

7.

Physical or Internal Level - it defines data-structures to store data and access methods used by the database. (how data is stored) Example: data is stored in the form of blocks of storage such as bytes, gigabytes etc.

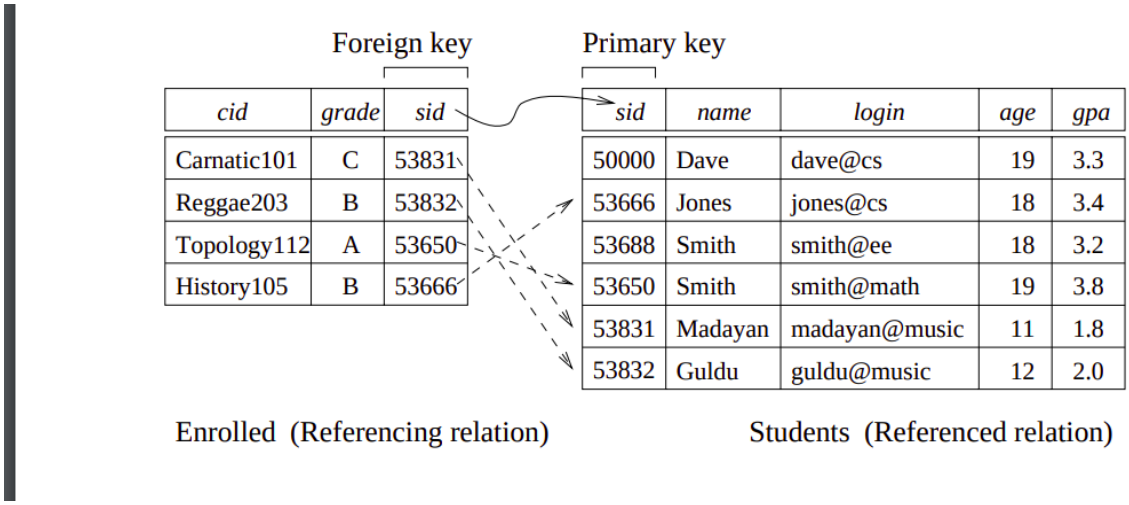
Logical or Conceptual Level – It defines what data is stored and describes the relationship among data.
Example: contains tables (fields and attributes) and relationships among table attributes.

View or External Level: every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database.

Physical level: describes how a record (e.g., instructor) is stored.
Logical level: describes data stored in database, and the relationships among the data.
View level: Views can hide information (such as an employee's salary) for security purposes.

2nd part of question 7
The lowest layer (i.e., the physical layer) of the database is the one which needs to be rewritten.
The disk space management layer has to be rewritten to take advantage of the new function on OS files.

8.



OBJ

Ans:

a)In this case the INSERT command is simply rejected.

b)The options are:

i)Delete all Enrolled rows that refer to the deleted Students row.

ii)Disallow the deletion of the Students row if an Enrolled row refers to it.

iii)Set the sid column to the sid of some (existing) 'default' student, for every Enrolled row that refers to the deleted Students row.

iv) For every Enrolled row that refers to it, set the sid column to null. In our example, this option conflicts with the fact that sid is part of the primary key of Enrolled and therefore cannot be set to null. Thus, we are limited to the first three options in our example, although this fourth option (setting the foreign key to null) is available in the general case.

c) The options here are similar to the previous case.

SQL-92 allows us to choose any of the four options on DELETE and UPDATE. For example, we can specify that when a Students row is deleted, all Enrolled rows that refer to it are to be deleted as well, but that when the sid column of a Students row is modified, this update is to be rejected if an Enrolled row refers to the modified Students row:

```
CREATE TABLE Enrolled
(
        sid CHAR(20),
        cid CHAR(20),
        grade CHAR(10),
        PRIMARY KEY (sid, cid),
        FOREIGN KEY (sid) REFERENCES Students ON DELETE CASCADE ON UPDATE NO
ACTION
)
```

The options are specified as part of the foreign key declaration. The default option is NO ACTION, which means that the action (DELETE or UPDATE) is to be rejected. Thus, the ON UPDATE clause in our

example could be omitted, with the same effect. The CASCADE keyword says that if a Students row is deleted, all Enrolled rows that refer to it are to be deleted as well. If the UPDATE clause specified CASCADE, and the sid column of a Students row is updated, this update is also carried out in each Enrolled row that refers to the updated Students row. If a Students row is deleted, we can switch the enrollment to a 'default' student by using ON DELETE SET DEFAULT. The default student is specified as part of the definition of the sid field in Enrolled; for example, sid CHAR(20) DEFAULT '53666'. Although the specification of a default value is appropriate in some situations (e.g., a default parts supplier if a particular supplier goes out of business), it is really not appropriate to switch enrollments to a default student. The correct solution in this example is to also delete all enrollment tuples for the deleted student (that is, CASCADE), or to reject the update. SQL also allows the use of null as the default value by specifying ON DELETE SET NULL.

9. Logical data independence means that users are shielded from changes in the logical structure of the data, while physical data independence insulates users from changes in the physical storage of the data.
Any change at the physical level, does not require to change at the application level. The change in the logical level requires a change at the application level.
Ex:
physical data independence - Change in compression techniques, Hashing algorithms and storage devices etc.
Logical data independence - Add/Modify or Delete a new attribute.

10. R1 to Rn are attributes of table R and S1 to Sn are the attributes of table S.

i) $\Pi_{R1,...,Rn}(\sigma_{R.1=S.1 \wedge ... \wedge R.n=S.n}(R \times S))$

ii) $\Pi_{R1,...,Rn}(R \bowtie S)$

11.

a) Select L from R1,R2,...,Rn where C
b) Select L from R1 natural join R2 natural join R3 … natural join Rn where C

12.

$R/S = \pi_F(R) - \pi_F((\pi_F(R) \times S) - R)$.