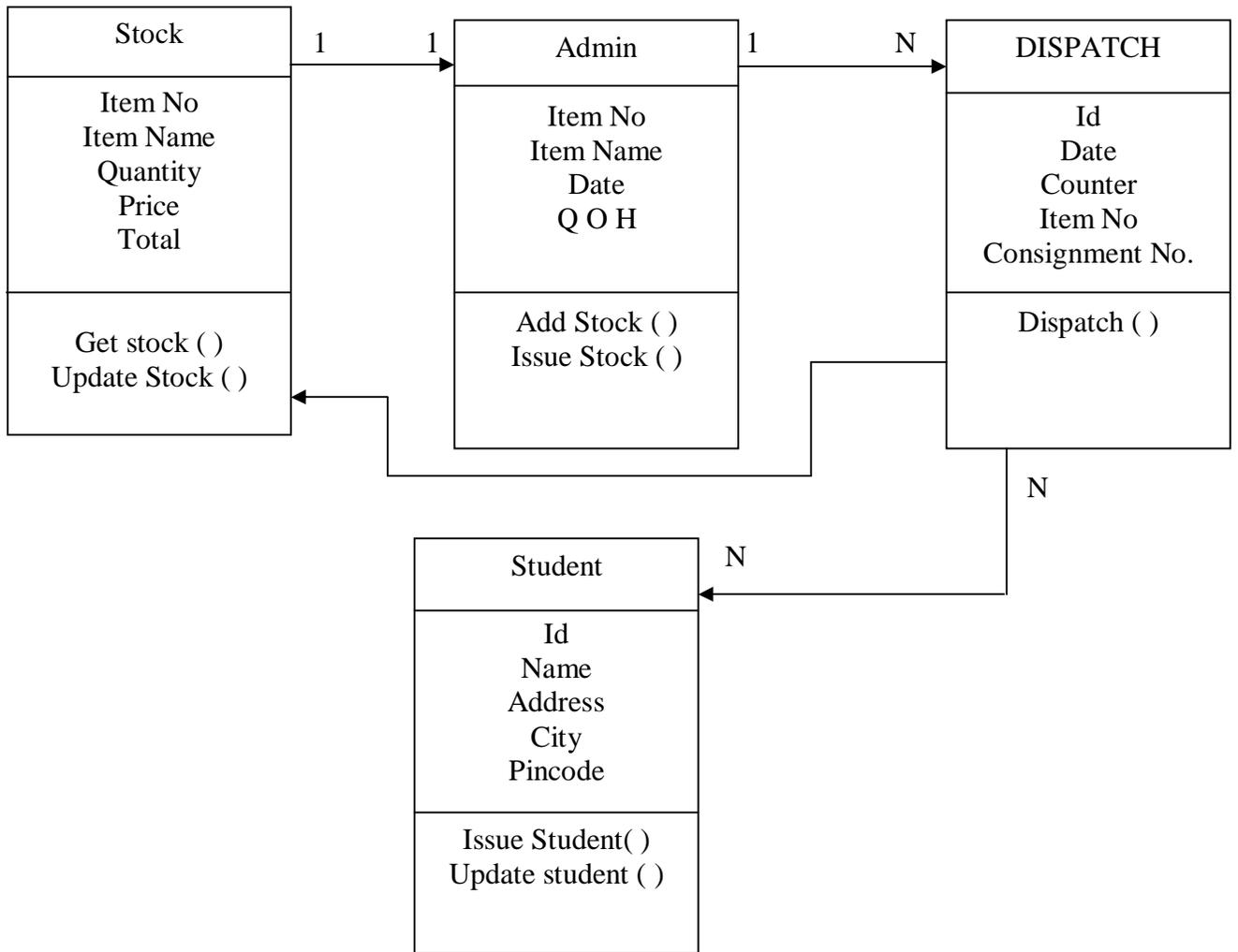


(b)



(C)
Question-1 (c)

Stock
Item No Item Name Quantity Price Total
Get stock () Update Stock ()

Admin
Item No Item Name Date Q O H
Add Stock () Issue Stock ()

DISPATCH
Id Date Counter Item No Consignment No.
Dispatch ()

Student
Id Name Address City Pincode
Issue Student() Update student ()

(d)

Select material, convert (material) from Stock
Group by Publication

(e)

Select material from stock

Question 2

Solution

(a) Select Supplier.su_id, supplier.Name.su (Supply of per quantity of supply) from supplier
On supplier.su_id inner join supply of par su_id group by su_id

(b) Begin Transaction Trans

```
Go
Use Adventure work 2008 R
GO
SELECT + Money
From Adventurework 2008 R
Human Resource.Account
Where Account No. = 123456
Commit Transaction @ Account No
Go
```

= Begin Trasaction Tran 2

```
Go
Use Adventure Work 2008 R2
Go
Insert into Adventure work 2008 R2
VALUES (10000, ACCOUNT No. )
Commit Transaction @ Account No.
```

(c) Create view supplier

As

Select * FROM Supplier

Where su_id = @ su_id

= create view system manager

As

Select * from supplier su_id, supplier.su_name

Supplier.su_address, supplier.su_status

Supply of part.su_id, supply of part. Pa_id, supply of part.

Date of supply, supply of part.quantity of

Supply, supply of part.su_price from

Supplier on supplier.su_id inner join

Supply of part.su_id

(d) Transaction processing is designed to maintain a computer system (typically a [database](#) or some modern filesystems) in a known, consistent state, by ensuring that any operations carried out on the system that are interdependent are either all completed successfully or all canceled successfully.

For example, consider a typical banking transaction that involves moving \$700 from a customer's savings account to a customer's checking account. This transaction is a single operation in the eyes of the bank, but it involves at least two separate operations in computer terms: debiting the savings account by \$700, and crediting the checking account by \$700. If the debit operation succeeds but the credit does not (or *vice versa*), the books of the bank will not balance at the end of the day. There must therefore be a way to ensure that either both operations succeed or both fail, so that there is never any inconsistency in the bank's database as a whole. Transaction processing is designed to provide this.

Deadlock Detection

Deadlock can occur whenever two or more processes are competing for limited resources and the processes are allowed to acquire and hold a resource (obtain a lock) thus preventing others from using the resource while the process waits for other resources.

Two common places where deadlocks may occur are with processes in an operating system (distributed or centralized) and with transactions in a database. The concepts discussed here are applicable to any system that allocates resources to processes.

Locking protocols such as the popular Two Phase Locking (see concurrency control) give rise to deadlock as follows: process A gets a lock on data item X while process B gets a lock on data item Y. Process A then tries to get a lock on Y. As Y is already locked, process A enters a blocked state. Process B now decides to get a lock

on X, but is blocked. Both processes are now blocked, and, by the rules of Two Phase Locking, neither will relinquish their locks.

This is a deadlock: process A is waiting for a resource held by process B and process B is waiting for a resource held by process A. No progress will take place without outside intervention. Several processes can be involved in a deadlock when there exists a cycle of processes waiting for each other. Process A waits for B which waits for C which waits for A.

Four conditions must hold for deadlock to occur:

1. Exclusive use - when a process accesses a resource, it is granted exclusive use of that resource.
1. 2. Hold and wait - a process is allowed to hold onto some resources while it is waiting for other resources.
2. 3. No preemption - a process cannot preempt or take away the resources held by another process.
3. 4. Cyclical wait - there is a circular chain of waiting processes, each waiting for a resource held by the next process in the chain.

QUESTION-3

(a) Select from Guide Table Group by Guide_Id

(b)

```
<University>
<Teacher>
<id> 101 </id>
<Name> Rajat Sharma </name>
<Qualification> B.Tech </Qualification>
<Work Experience> 3 Years </work experience>
<published materials> Data Structure </published materials>
</Teacher>
</University>
```

=Begin Trasaction Trans 3

Go

Use Adsventrue Work 208 R2

Go

Update Adventure work 2008 R2 Set

Interest = @ interest where

Account No. = @ Account No.

Commit Transaction @ Acocount, @ Interest

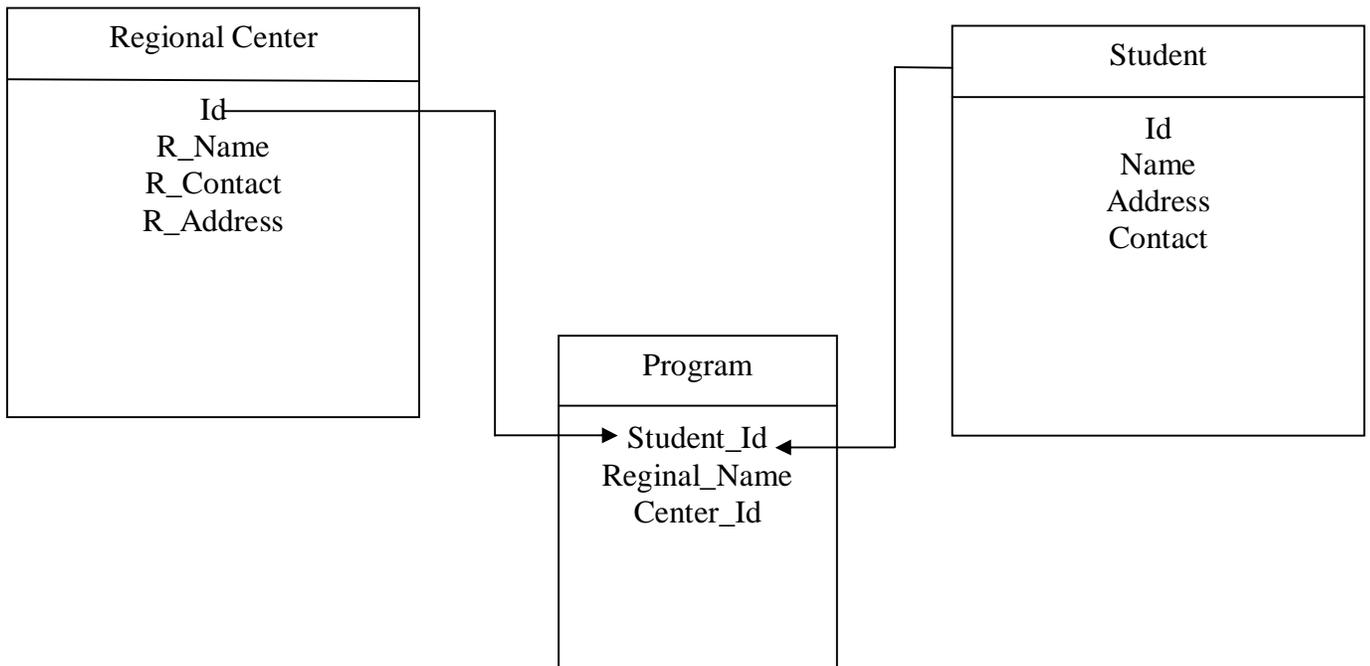
GO

= Begin Trasaction Trans 4

Go

Use adventure Work 208 R2
Go
Select * from Adventure work 2008 Rs
Commit Transaction
Go.

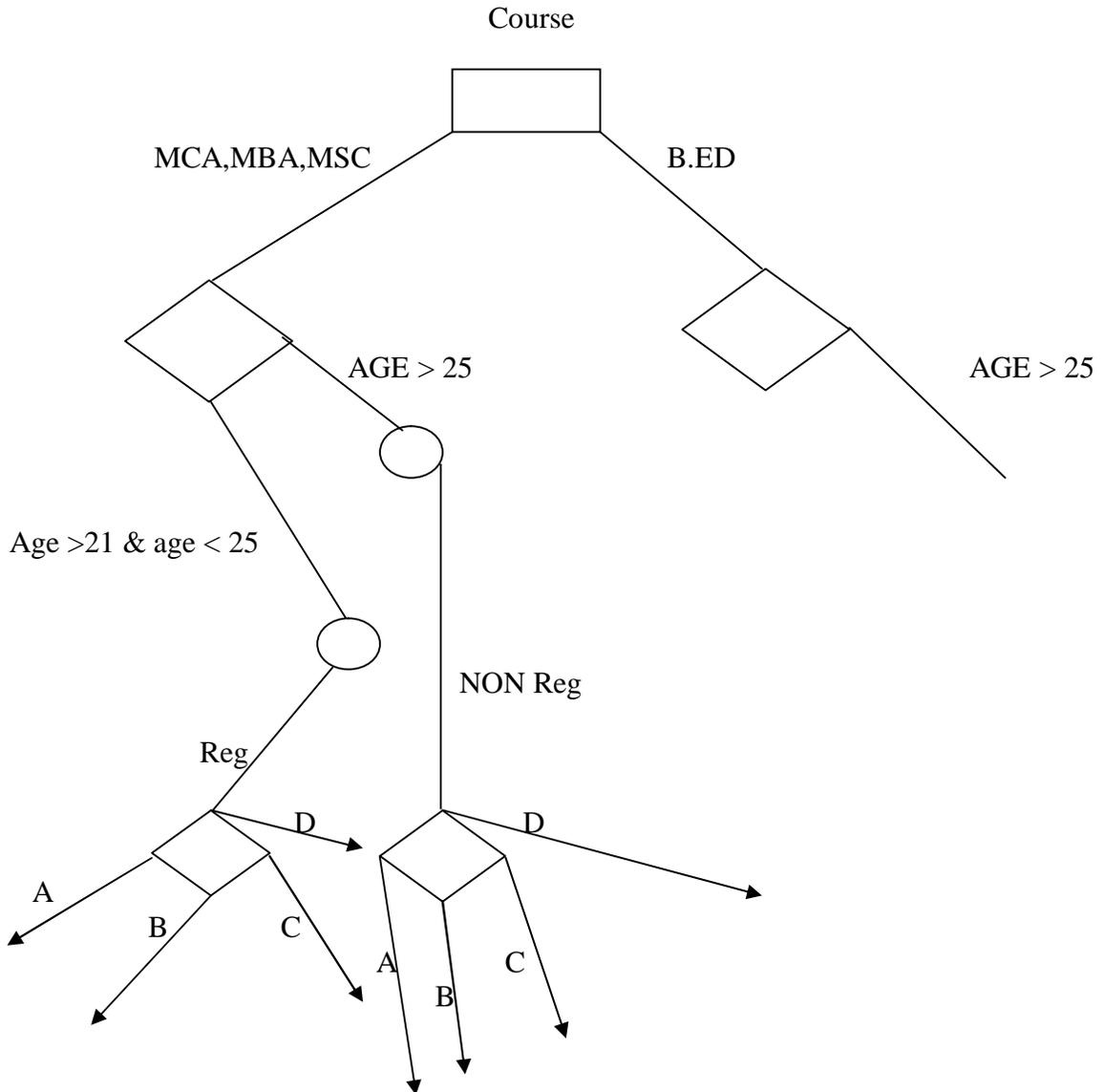
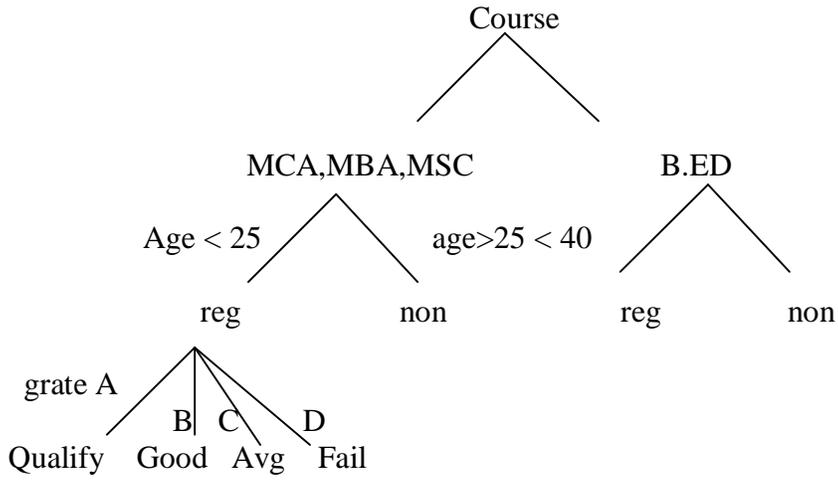
(c)

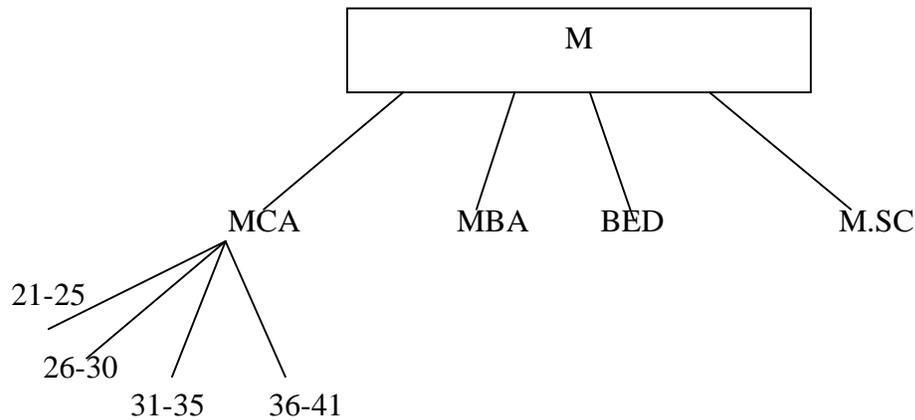


Star Schema

```
Select Regional Center_Id Regional Center.Name, count(student.Id)
From Regional Center 'inner join Student on Regional Center.Id=Student_Id group by
Regional.Id
```

(d)





Question 4)

a)i)

Ans.

A **knowledge base** (abbreviated **KB**, **kb** or $\Delta^{[1][2]}$) is a special kind of database for knowledge management, providing the means for the computerized collection, organization, and retrieval of knowledge. Also a collection of data representing related experiences, their results are related to their problems and solutions.

The knowledge based systems are artificial intelligent tools working in a narrow domain to provide intelligent decisions with justification. Knowledge is acquired and represented using various knowledge representation techniques rules, frames and scripts. The basic advantages offered by such system are documentation of knowledge, intelligent decision support, self learning, reasoning and explanation

Question 4)a)ii)

Ans.

A **mobile database** is a database that can be connected to by a mobile computing device over a mobile network. The client and server have wireless connections. A cache is maintained to hold frequent data and transactions so that they are not lost due to connection failure. A database is a structured way to organize information. This could be a list of contacts, price information or distance travelled.^[1]

The use of laptops, mobiles and PDAs is increasing and likely to increase in the future^[citation needed] with more and more applications residing in the mobile systems. While those same analysts can't tell us exactly which applications will be the most popular, it is clear that a large percentage will require the use of a database of some sort. Many applications such as databases would require the ability to download information from an information repository and operate on this information even when out of range or disconnected.

An example of this is a mobile workforce. In this scenario user would require to access and update information from files in the home directories on a server or customer records from a database. This type of access and work load generated by such users is different from the traditional workloads seen in client-server systems of today. With the advent of mobile databases, now users can load up their smart phones or PDAs with mobile databases to exchange mission-critical data remotely without worrying about time or distance. Mobile databases let employees enter data on the fly. Information can be synchronized with a server database at a later time.

Question 4)a)iii)

Ans.

A **digital library** is a library in which collections are stored in digital formats (as opposed to print, microform, or other media) and accessible by computers.^[1] The digital content may be stored locally, or accessed remotely via computer networks. A digital library is a type of information retrieval system.

The *DELOS Digital Library Reference Model*^[2] defines a digital library as:

An organization, which might be virtual, that comprehensively collects, manages and preserves for the long term rich digital content, and offers to its user communities specialized functionality on that content, of measurable quality and according to codified policies.

Question 4)a)iv)

Ans.

A **spatial database** is a database that is optimized to store and query data related to objects in space, including points, lines and polygons. While typical databases can understand various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types. These are typically called *geometry* or *feature*. The Open Geospatial Consortium created the Simple Features specification and sets standards for adding spatial functionality to database systems.^[1]

Question 4) b) i).

Ans.

An **index** is a performance-tuning method of allowing faster retrieval of records. An index creates an entry for each value that appears in the indexed columns. By default, Oracle creates B-tree indexes.

Create an Index

The syntax for creating a index is:

```
CREATE [UNIQUE] INDEX index_name  
ON table_name (column1, column2, . column_n)  
[ COMPUTE STATISTICS ];
```

Question 4) b) ii).

Ans.

By default, PostgreSQL's isolation mode is set to `READ COMMITTED`, for the `Read Committed` level, as listed in Table 9-7. The other mode available is `SERIALIZABLE`, for the `Serializable` level.

At the time of writing, PostgreSQL does not implement the intermediate level `Repeatable Read` or the entry level `Read Uncommitted`. Generally, `Read Uncommitted` is such poor behavior that few databases offer it as an option, and it would be a rare application that was brave (or foolhardy!) enough to choose to use it. The intermediate level `Repeatable Read` provides added protection only against phantom reads, which are extremely rare, so the lack of this level is of no real consequence. It is common for databases to offer less than the full set of possibilities, and providing `Read Committed` and `Serializable` is a good compromise solution.

You can change the isolation level by using the `SET TRANSACTION ISOLATION LEVEL` command, which has the following syntax:

```
SET TRANSACTION ISOLATION LEVEL { READ COMMITTED | SERIALIZABLE }
```

Unless you have a very good reason to change it, we suggest you don't adjust the default isolation level of your PostgreSQL database.

Question 4) b) iii).

Ans.

`ctid`

The physical location of the row version within its table. Note that although the `ctid` can be used to locate the row version very quickly, a row's `ctid` will change each time it is updated or moved by `VACUUM FULL`. Therefore `ctid` is useless as a long-term row identifier. The `OID`, or even better a user-defined serial number, should be used to identify logical rows.

Question 4) b) iv).

Ans.

The *Server Programming Interface* (SPI) gives users the ability to run SQL queries inside user-defined C functions. The available Procedural Languages (PL) give an alternate means to access these capabilities.

In fact, SPI is just a set of native interface functions to simplify access to the Parser, Planner, Optimizer and Executor. SPI also does some memory management.

To avoid misunderstanding we'll use *function* to mean SPI interface functions and *procedure* for user-defined C-functions using SPI.

SPI procedures are always called by some (upper) Executor and the SPI manager uses the Executor to run your queries. Other procedures may be called by the Executor running queries from your procedure.

Note, that if during execution of a query from a procedure the transaction is aborted then control will not be returned to your procedure. Rather, all work will be rolled back and the server will wait for the next command from the client. This will be changed in future versions.

Other restrictions are the inability to execute BEGIN, END and ABORT (transaction control statements) and cursor operations. This will also be changed in the future.

Question 4) b) v).

Ans. PostgreSQL's *statistics collector* is a subsystem that supports collection and reporting of information about server activity. Presently, the collector can count accesses to tables and indexes in both disk-block and individual-row terms. It also supports determining the exact command currently being executed by other server processes.

Question 4) c) i).

Ans.

A system global area (SGA) is a group of shared memory structures that contain data and control information for one Oracle database instance. If multiple users are concurrently connected to the same instance, then the data in the instance's SGA is shared among the users. Consequently, the SGA is sometimes called the shared global area.

An SGA and Oracle processes constitute an Oracle instance. Oracle automatically allocates memory for an SGA when you start an instance, and the operating system reclaims the memory when you shut down the instance. Each instance has its own SGA.

The SGA is read/write. All users connected to a multiple-process database instance can read information contained within the instance's SGA, and several processes write to the SGA during execution of Oracle.

Question 4) c) ii).

Ans.

The control files of a database store the status of the physical structure of the database. The control file is absolutely crucial to database operation. It contains (but is not limited to) the following types of information:

- Database information (RESETLOGS SCN and their time stamp)
- Archive log history
- Tablespace and datafile records (filenames, datafile checkpoints, read/write status, offline or not)
- Redo threads (current online redo log)
- Database's creation date
- database name
- current archive log mode
- Log records (sequence numbers, SCN range in each log)
- RMAN catalog
- Database block corruption information
- Database ID, which is unique to each DB

Question 4) c) iii).

Ans.

Use the `CREATE CLUSTER` statement to create a cluster. A **cluster** is a schema object that contains data from one or more tables, all of which have one or more columns in common. Oracle Database stores together all the rows from all the tables that share the same cluster key.

For information on existing clusters, query the `USER_CLUSTERS`, `ALL_CLUSTERS`, and `DBA_CLUSTERS` data dictionary views.

Question 4) c) iv).

Ans.

Recovery manager is a platform independent utility for coordinating your backup and restoration procedures across multiple servers. In my opinion it's value is limited if you only have one or two instances, but it comes into it's own where large numbers of instances on multiple platforms are used. The reporting features alone mean that you should never find yourself in a position where your data is in danger due to failed backups.

Question 4) c) v).

Ans.

A materialized view is a database object that contains the results of a query. They are local copies of data located remotely, or are used to create summary tables based on aggregations of a table's data. Materialized views, which store data based on remote tables are also, know as snapshots.

A materialized view can query tables, views, and other materialized views. Collectively these are called master tables (a replication term) or detail tables (a data warehouse term).

For replication purposes, materialized views allow you to maintain copies of remote data on your local node. These copies are read-only. If you want to update the local copies, you have to use the Advanced Replication feature. You can select data from a materialized view as you would from a table or view.

Visit Ignou4u.blogspot.com for more Solved Assignments

For data warehousing purposes, the materialized views commonly created are aggregate views, single-table aggregate views, and join views.

In this article, we shall see how to create a Materialized View and discuss Refresh Option of the view.

In replication environments, the materialized views commonly created are primary key, rowid, and subquery materialized views.