Q1)

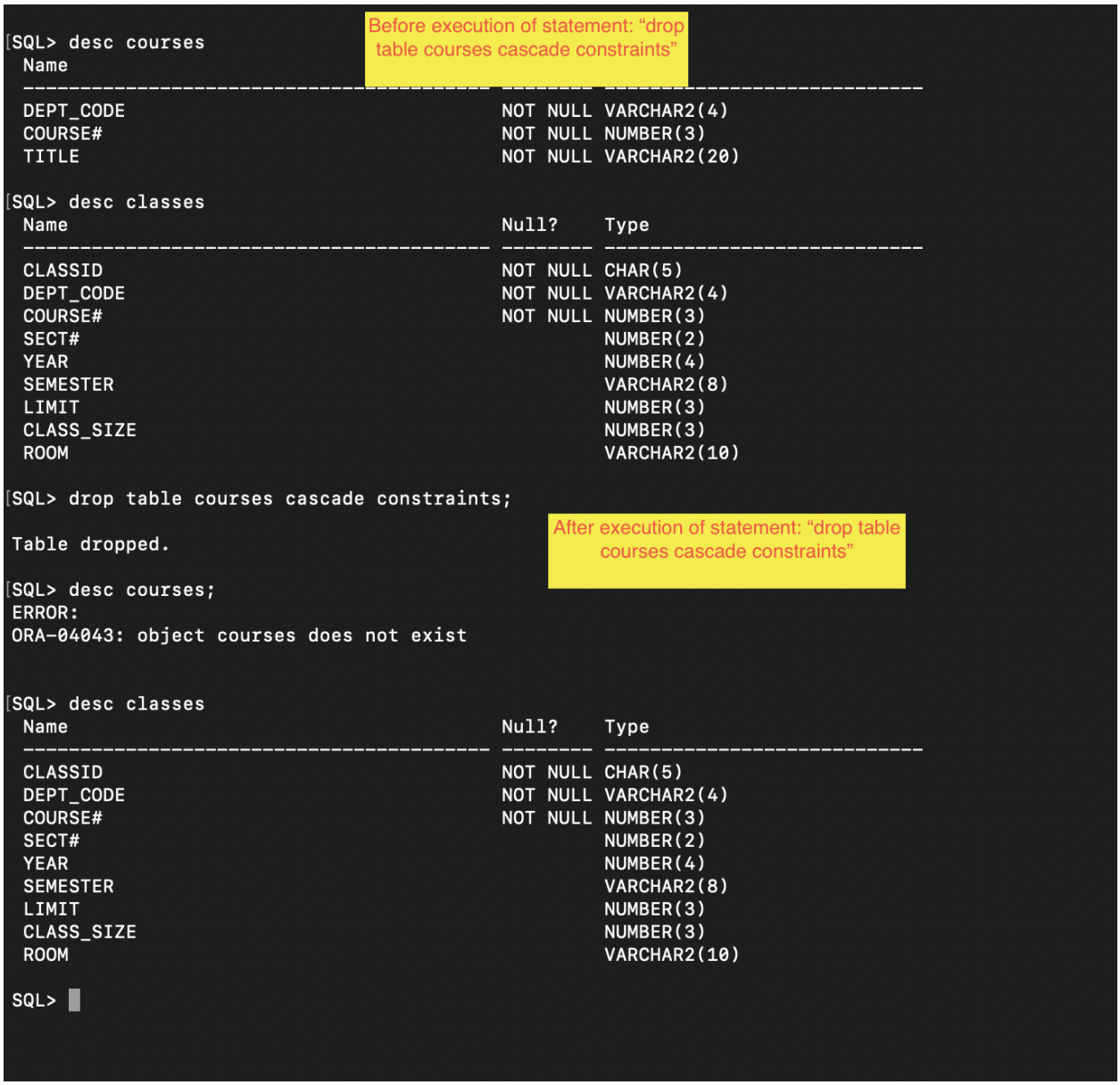
Actual SQL statement: drop table courses;

Returned Message:

drop table courses
*****
ERROR at line 1:
ORA-02449: unique/primary keys in table referenced by foreign keys

Purpose of the message The ORA-02449 error states that "unique/primary keys in a table referenced by foreign keys" happens when we seek to remove a table with foreign key constraints referencing the table we want to dump. This means that in order to maintain referential integrity while dumping the table, we must first make sure that any foreign key constraints have been removed.

b)


Before execution of statement: “drop table courses cascade constraints”
SQL> desc courses
Name Null? Type
DEPT_CODE NOT NULL VARCHAR2(4)
COURSE# NOT NULL NUMBER(3)
TITLE NOT NULL VARCHAR2(20)
SQL> desc classes
Name Null? Type
CLASSID NOT NULL CHAR(5)
DEPT_CODE NOT NULL VARCHAR2(4)
COURSE# NOT NULL NUMBER(3)
SECT# NUMBER(2)
YEAR NUMBER(4)
SEMESTER VARCHAR2(8)
LIMIT NUMBER(3)
CLASS_SIZE NUMBER(3)
ROOM VARCHAR2(10)
SQL> drop table courses cascade constraints;
Table dropped.
After execution of statement: “drop table courses cascade constraints”
SQL> desc courses;
ERROR:
ORA-04043: object courses does not exist
SQL> desc classes
Name Null? Type
CLASSID NOT NULL CHAR(5)
DEPT_CODE NOT NULL VARCHAR2(4)
COURSE# NOT NULL NUMBER(3)
SECT# NUMBER(2)
YEAR NUMBER(4)
SEMESTER VARCHAR2(8)
LIMIT NUMBER(3)
CLASS_SIZE NUMBER(3)
ROOM VARCHAR2(10)
SQL>

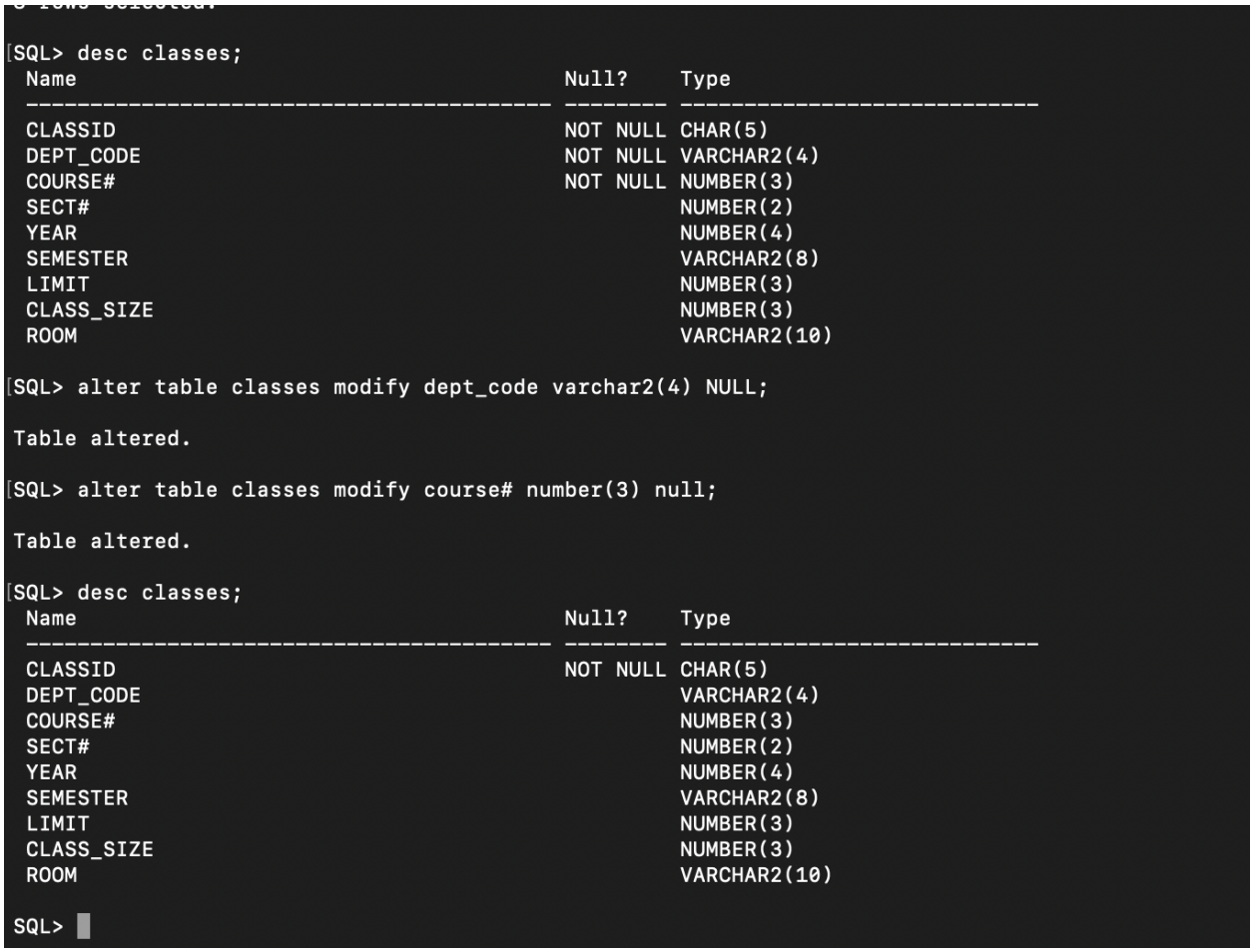
```
[SQL> select * from classes;

CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0001 CS          432          1       2021 Spring           13         12
LH 005

c0002 Math        314          1       2020 Fall             13         12
LH 009

c0003 Math        314          2       2020 Fall             14         11
LH 009


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0004 CS          432          1       2020 Spring           13         13
SW 222

c0005 CS          536          1       2021 Spring           14         13
LH 003

c0006 CS          532          1       2021 Spring           10          9
LH 005


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0007 CS          550          1       2021 Spring           11         11
WH 155


7 rows selected.

[SQL> select * from courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          432 database systems
Math        314 discrete math
CS          240 data structure
CS          575 algorithms
CS          532 database systems
CS          550 operating systems
CS          536 machine learning

7 rows selected.

[SQL> drop table courses cascade constraints;

Table dropped.

[SQL> select * from courses;
select * from courses
              *
ERROR at line 1:
ORA-00942: table or view does not exist


[SQL> select * from classes;

CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0001 CS          432          1       2021 Spring           13         12
LH 005

c0002 Math        314          1       2020 Fall             13         12
LH 009

c0003 Math        314          2       2020 Fall             14         11
LH 009


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0004 CS          432          1       2020 Spring           13         13
SW 222

c0005 CS          536          1       2021 Spring           14         13
LH 003

c0006 CS          532          1       2021 Spring           10          9
LH 005


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0007 CS          550          1       2021 Spring           11         11
WH 155


7 rows selected.
```

Before execution of statement: “drop table courses cascade constraints”

After execution of statement: “drop table courses cascade constraints”

i) Yes, the courses table got dropped after execution of the statement “drop table courses cascade constraints” , as we can see in the screenshot of the terminal above, there are two texts which show the tables description before and after the execution of the statement.
ii) Also, none of the tuples from classes have been deleted, as we can see from the second screenshot of the terminal.
iii) Reasons why courses table dropped on the command : “drop table courses cascade constraints;”, because this command will first of all drop the table courses compulsorily and the constraints that it is associated with, which will also include the foreign key constraints between courses and classes if any. So, foreign key constraints need to be dropped before dropping the courses table, and the keyword “cascade” specifies that all related constraints are dropped as well.
iv) “On delete cascade” for classes did not activate when the courses table got dropped, it only dropped the key constraints between the two tables courses and classes, and did not affect any record from the courses table.

Q2) **Classes table has been altered for the question 2, as we can see, before writing the alter command for classes, dept_code and course# will not take null values, but after entering the command, they will, we can see from the description.**



```
SQL> desc classes;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASSID                                   NOT NULL CHAR(5)
 DEPT_CODE                                 NOT NULL VARCHAR2(4)
 COURSE#                                   NOT NULL NUMBER(3)
 SECT#                                              NUMBER(2)
 YEAR                                               NUMBER(4)
 SEMESTER                                           VARCHAR2(8)
 LIMIT                                              NUMBER(3)
 CLASS_SIZE                                         NUMBER(3)
 ROOM                                               VARCHAR2(10)

SQL> alter table classes modify dept_code varchar2(4) NULL;

Table altered.

SQL> alter table classes modify course# number(3) null;

Table altered.

SQL> desc classes;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASSID                                   NOT NULL CHAR(5)
 DEPT_CODE                                          VARCHAR2(4)
 COURSE#                                            NUMBER(3)
 SECT#                                              NUMBER(2)
 YEAR                                               NUMBER(4)
 SEMESTER                                           VARCHAR2(8)
 LIMIT                                              NUMBER(3)
 CLASS_SIZE                                         NUMBER(3)
 ROOM                                               VARCHAR2(10)

SQL>
```

a) **Tables courses and classes with their respective records:**

```
SQL> select * from courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          432 database systems
Math        314 discrete math
CS          240 data structure
CS          575 algorithms
CS          532 database systems
CS          550 operating systems
CS          536 machine learning

7 rows selected.

SQL> select * from classes;

CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0001 CS          432          1       2021 Spring           13         12
LH 005

c0002 Math        314          1       2020 Fall             13         12
LH 009

c0003 Math        314          2       2020 Fall             14         11
LH 009


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0004 CS          432          1       2020 Spring           13         13
SW 222

c0005 CS          536          1       2021 Spring           14         13
LH 003

c0006 CS          532          1       2021 Spring           10          9
LH 005


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0007 CS          550          1       2021 Spring           11         11
WH 155


7 rows selected.

SQL>
```

b)

```
SQL> insert into classes values('C001', 'D001', 123, 1, 2023, 'Spring', 30, 0, 'Room 101');
insert into classes values('C001', 'D001', 123, 1, 2023, 'Spring', 30, 0, 'Room 101')
*
ERROR at line 1:
ORA-02290: check constraint (NMASAND1.SYS_C00542152) violated


SQL>
```

We can see, from the error message, we tried inserting values that do not match anything in the courses table, and this the output we receive, from the terminal it is evident, that the system is enforcing referential integrity constraint.

c) We tried inserting two null values using the query Insert into classes values('c0002', NULL, NULL, 1, 2023, 'Spring', 30, 0, 'LH 155'), but this failed as the constraints are getting violated, even after altering classes

```
SQL> insert into classes values('c0002', NULL, NULL, 1, 2023, 'Spring', 30, 0, 'LH 155');
insert into classes values('c0002', NULL, NULL, 1, 2023, 'Spring', 30, 0, 'LH 155')
*
ERROR at line 1:
ORA-02290: check constraint (NMASAND1.SYS_C00542155) violated


SQL>
```

d) We tried inserting non null value for dept_code and null value for course# , but the insertion failed as foreign key constraints between courses and classes are still in place

```
SQL> insert into classes values('C003', 'D002', NULL, 1, 2023, 'Spring', 30, 0, 'Room 103');
insert into classes values('C003', 'D002', NULL, 1, 2023, 'Spring', 30, 0, 'Room 103')
*
ERROR at line 1:
ORA-02290: check constraint (NMASAND1.SYS_C00542152) violated


SQL>
```

e) Also, insertion failed for inserting NULL value for dept_code and Non Null value for course#

```
SQL> insert into classes values('C004', NULL, 234, 1, 2023, 'Spring', 30, 0, 'Room 104');
insert into classes values('C004', NULL, 234, 1, 2023, 'Spring', 30, 0, 'Room 104')
*
ERROR at line 1:
ORA-02290: check constraint (NMASAND1.SYS_C00542152) violated


SQL>
```

f) Insertion failed when we tried inserting into classes with a null value for dept code and non null value for course# (the value not being in course), here also referential integrity constraint is violated

```
SQL> insert into classes values('c0009', NULL, 999, 1, 2023, 'Spring', 30, 0, 'LH 535');
insert into classes values('c0009', NULL, 999, 1, 2023, 'Spring', 30, 0, 'LH 535')
*
ERROR at line 1:
ORA-02290: check constraint (NMASAND1.SYS_C00542155) violated


SQL>
```

e) Content of classes:
All attempts to insert latest records failed

```
SQL> select * from classes;

CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0001 CS          432          1       2021 Spring           13         12
LH 005

c0002 Math        314          1       2020 Fall             13         12
LH 009

c0003 Math        314          2       2020 Fall             14         11
LH 009


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0004 CS          432          1       2020 Spring           13         13
SW 222

c0005 CS          536          1       2021 Spring           14         13
LH 003

c0006 CS          532          1       2021 Spring           10          9
LH 005


CLASS DEPT    COURSE#      SECT#       YEAR SEMESTER      LIMIT CLASS_SIZE
----- ---- ---------- ---------- ---------- -------- ---------- ----------
ROOM
----------
c0007 CS          550          1       2021 Spring           11         11
WH 155


7 rows selected.

SQL>
```

f) None of the mentioned rules of inserting (null, null) values and (null, NON null) values are followed by Oracle, as we can see the output answer to the questions asked above. So, we can conclude that when a foreign key constraint contains a null value, oracle will always consider the constraint is violated, no matter the insertion be (null, null) or (a, null) or (null, a)

Q3)

```
SQL> create view CS_Courses as
  2  select * from courses
  3  where dept_code='CS';
```

View created.

SQL> /* a) */

```
SQL> insert into CS_Courses values('CS', 537, 'TDS');

1 row created.
```

```
SQL> select * from CS_Courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          240 data structure
CS          432 database systems
CS          532 database systems
CS          536 machine learning
CS          537 TDS
CS          550 operating systems
CS          575 algorithms

7 rows selected.

SQL> /* before deletion */
SQL> select * from CS_Courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          240 data structure
CS          432 database systems
CS          532 database systems
CS          536 machine learning
CS          537 TDS
CS          550 operating systems
CS          575 algorithms

7 rows selected.

SQL> delete from CS_Courses where DEPT_CODE='CS' AND COURSE#=537;

1 row deleted.

SQL> /* after deletion */
SQL> select * from CS_Courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          240 data structure
CS          432 database systems
CS          532 database systems
CS          536 machine learning
CS          550 operating systems
CS          575 algorithms
```

6 rows selected.

⇒ We were successful in insertion of a CS course into view CS_Courses and were also able to delete the same, as shown from the above spooled text

SQL> /* b) */
SQL> /* Here we are entering the course that is logically not a part of CS_Courses view */
SQL> insert into CS_Courses values('PHY', 569, 'GRAVITY');

1 row created.
SQL> /* after insertion attempt */

SQL> select * from CS_Courses;

DEPT COURSE# TITLE
---- ---------- --------------------
CS 240 data structure
CS 432 database systems
CS 532 database systems
CS 536 machine learning
CS 550 operating systems
CS 575 algorithms

6 rows selected.

SQL> /* insertion is not successful */

SQL> delete from CS_Courses where DEPT_CODE='PHY' AND COURSE#=569;

0 rows deleted.
SQL> spool off

⇒ Finally, for quest 3b), we can see from above spooled text, that if we try inserting a course, that logically does not belong to CS_Courses view, we can see that we were not successful in insertion of such a record into the view

```
SQL> insert into CS_Courses values('PHY', 569, 'GRAVITY');

1 row created.

SQL> select * from CS_Courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          240 data structure
CS          432 database systems
CS          532 database systems
CS          536 machine learning
CS          550 operating systems
CS          575 algorithms

6 rows selected.

SQL> /* insertion is not successful */
SQL> delete from CS_Courses where DEPT_CODE='PHY' AND COURSE#=569;

0 rows deleted.

SQL>
```

The screenshot shows that the row was created, it is right but they go into main Courses table, not the view, as we can see from the screenshot below.

```
SQL> select * from courses;

DEPT    COURSE# TITLE
---- ---------- --------------------
CS          432 database systems
Math        314 discrete math
CS          240 data structure
CS          575 algorithms
CS          532 database systems
CS          550 operating systems
CS          536 machine learning
MATH        101 Calculus
PHY         569 GRAVITY

9 rows selected.

SQL>
```

Q4)
The main distinction between a primary index and a secondary index is that a primary index can only exist once, whereas further indexes must be secondary indexes. Assuming that an attribute A should have a main index if searching on it will produce more records or if A is the primary key for the specified relation.
So, for now, we will consider a relation “Students” which has attributes of “Roll No”, “Name”, “Department”, and “GPA”. Now, we fire queries to extract the information of a specific student depending on their roll numbers.

So, if a primary index is formed on the “Roll_No” attribute, the data will be adjusted in the relation “Students” based on their “Roll_No”. The data will be in a sequence depending on the Roll Nos. So, when we fire queries on student relation, the system here “oracle” will be able to search and mark the location of the required record from relation “Students”.
This is because if records of a relation R are stored in ascending or descending order, then only the index is a primary index.

If a secondary index is created on “Roll_No”, then according to the definition of secondary index, values based on “Roll_No” attribute will not be sorted. So, when we will fire a query on “Students” relation based on “Roll_No” attribute, the system will take time to locate the desired records in the secondary index and then will have to point out actual records.

So, heading to conclusion, we can say that having a secondary index on “Roll_No” attribute will result in higher I/O costs compared to using a primary index because then the oracle will have to get access to two different data structures, the secondary index and then the “Students” relation to fetch the required records.

Therefore, if we use a primary index on the “Roll_No” attribute it can result in a better performance and lower I/O cost.

Q5)