CS 432: Databases

Assignment 2: DEVELOPING THE DBMS

Topic: Placement Management

Group Members

G1:
Dhyeykumar Thummar - 20110059
Ksheer Sagar Agrawal - 20110098
Saatvik Rao - 20110175
Sanskar Sharma - 20110185
Mihir Sutariya - 20110208
Utkarsh Mishra - 20110218

G2:
Pushpendra Pratap Singh - 20110151
R Yeeshu Dhurandhar - 20110152
Sahil Agrawal - 20110178
Shruhrid Banthia - 20110198
Tanvi Dixit - 20110212

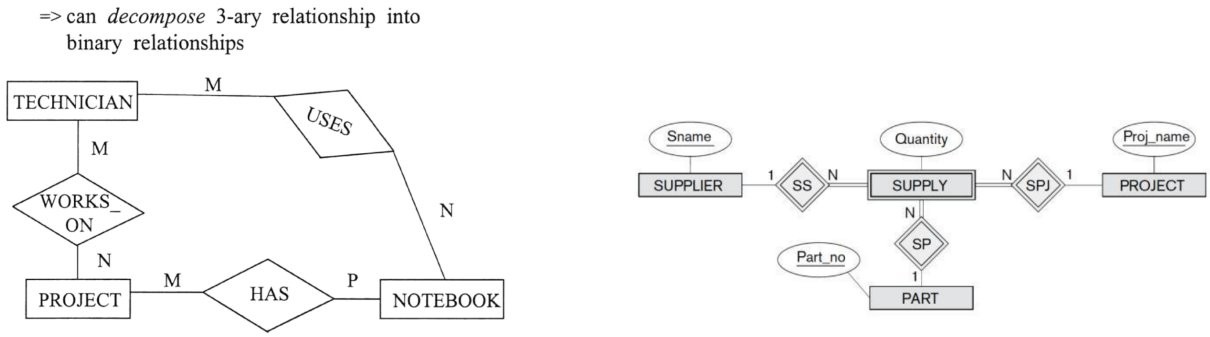
3.1 Responsibility of G1:

1. Populate the tables that you created in the previous assignment with random data with the following constraints. All tables must follow the ACID properties and the previous constraints mentioned in Assignment 1.
2. Please explain and implement the indexing over one of the columns (where the search needs to be optimized), user-defined data types, and table extensions.

Following Assignment 1, we made a few modifications, including but not limited to breaking down schemas and schema redesign.

Some of the major changes are as follows:

Two ways to break ternary relationship:



Source: StackOverflow

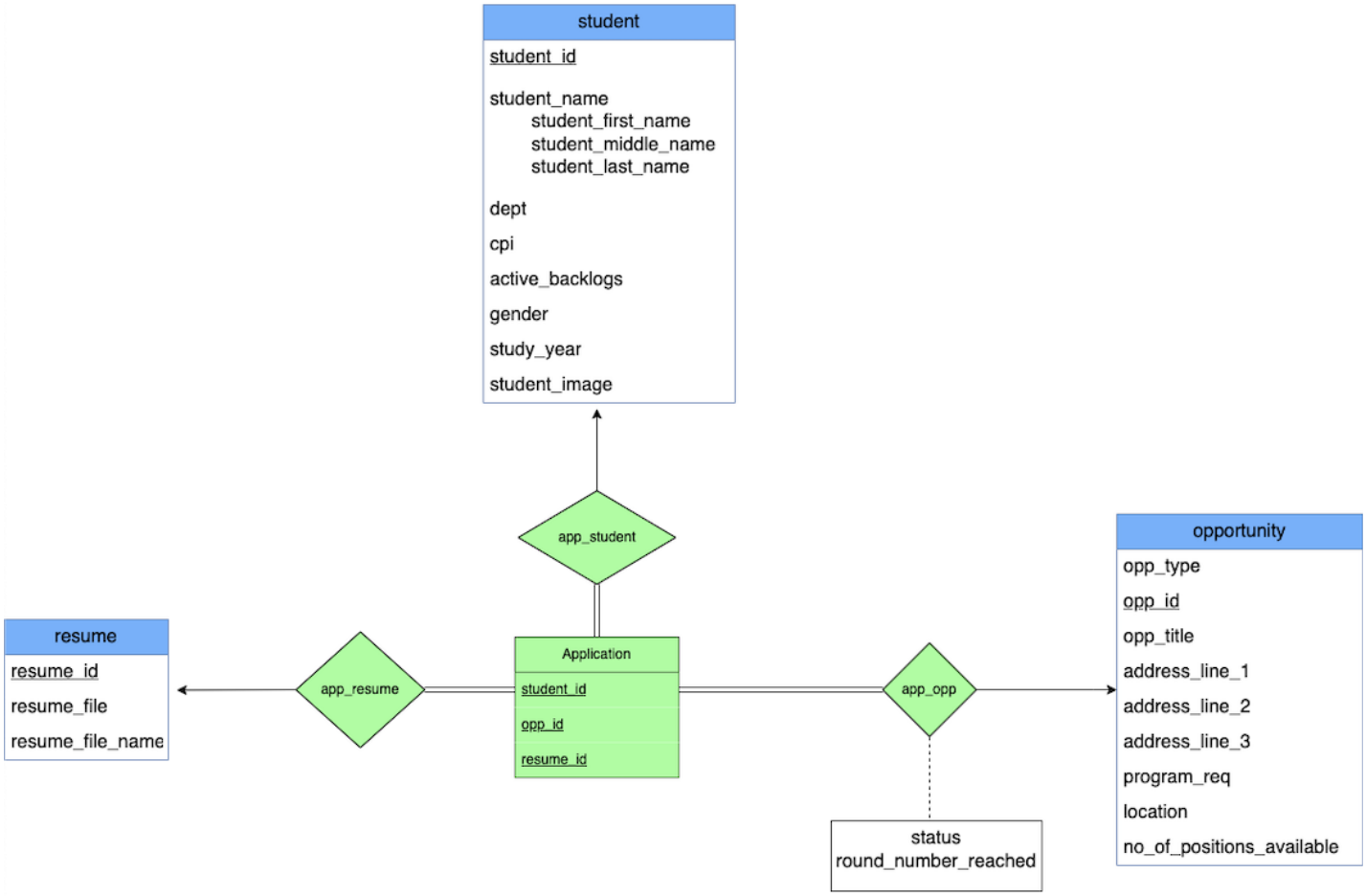
The first one is:

- Space Efficient
- Time Inefficient

The second one is:

- Space Inefficient
- Time Efficient

We used the second method for breaking our ternary relationship into 3 binary relationships. This helped us better express the cardinalities in our database.



Optimizing for queries to run on the opportunity table:

We also broke down the opportunity table into two parts - opportunity and requirements. This was done for removing redundancy. According to our previous schema, when a company floats an opportunity, if it allows both Bachelors and Masters students to apply, we would have to have 2 tuples, one for bachelors and one for masters for the same opportunity. This is because when we add the requirements about the year, we would need two tuples because an opportunity might open up for Masters 2nd year and Bachelors 2nd year.

Now that we broke down the table, there would be a single tuple in the opportunity table along with different tuples for different requirements that an opportunity might have.

Please explain and implement the indexing over one of the columns (where the search needs to be optimized)

Indexing Based on most unique and most commonly used attributes

Indexing has been done on schemas in which queries are made extensively. Indexing is done to reduce the query time and improve the efficiency of our database system. In our database we used a set of commonly queried attributes as our indices.

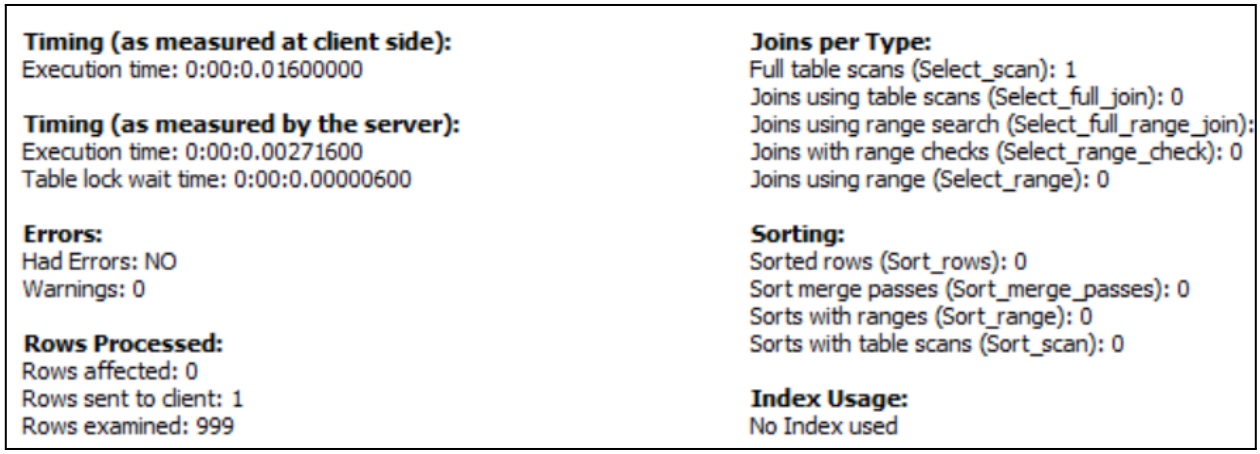
How does indexing help?

The set of attributes on which indexing is defined is used to map all the rows in the schema, So rather than searching the entire n rows, indexing maps rows based on certain attributes used to define indexing.

1. Company: **company_name**
   Many companies will have somewhat unique name.It is the most searched attribute. It is unique.
2. Opportunity: **opp_title**
   Opp_title describes the job role. It is one of the most searched and most unique attributes in opportunity schema.
3. Point_of_contact: We know the opportunity. Since queries would be mostly company specific and knowing opportunity, indexing is done on opp_id.
4. Requirement: **status,min_cpi_req,dept_req,program_req** would reduce a lot of searching. Can't do indexing on deadline otherwise index size will be very large. Status, min_cpi and program_req are attributes with less values. So the size of the index table will be small. Also these are commonly searched attributes.
5. Selection_procedure: **round_number,round_type.**
   Selection procedure will contain entries of different opportunities. So for each opportunity we will have a certain round_number and round_type. These are commonly searched attributes and should reduce search time.
6. Student: **first_name,middle_name,last_name**
   First_name,middle_name,last_name are commonly searched attributes and fairly unique.
   Students can have the same first_name and the same middle_name.
7. Placement: **company_name.**
   Many students will be selected in many companies over time.Company_name is a commonly searched attribute and is unique.
8. Internship: **company_name.** Adding medium won't make much difference index search will become double.
   Many students will be selected in many companies over time.Company_name is a commonly searched attribute and is unique.
9. Application: **student_id.** (best possible)
   Application is a schema with a large size.Student apply in different companies for placement and internship and with different resume. So for indexing student_id is used. It is unique and reduces search space.
10. App_opp: **student_id, status.** Adding status would reduce search space by almost half.
    There can be multiple entries of students with status: ‘eligible','not eligible','selected','rejected' for different opportunities.Application is a schema with a large size. So for indexing student_id is used. It is unique and reduces search space.

Indexing Query Statistics

1. **SELECT * from student where student_first_name = 'jikiv' and student_middle_name='szkha' and student_last_name='ghsyb';**



CREATE INDEX student_index ON student (student_first_name,student_middle_name,student_last_name);
SELECT * from student where student_first_name = 'jikiv' and student_middle_name='szkha' and student_last_name='ghsyb';

Timing (as measured at client side):
Execution time: 0:00:0.00000000

Timing (as measured by the server):
Execution time: 0:00:0.00125670
Table lock wait time: 0:00:0.00000700

Errors:
Had Errors: NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 1

Joins per Type:
Full table scans (Select_scan): 0
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0

Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0

Index Usage:
At least one Index was used

2. **SELECT * from opportunity where opp_title = 'Software Development Internship';**

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.00000000

Timing (as measured by the server):
Execution time: 0:00:0.00073130
Table lock wait time: 0:00:0.00000600

Errors:
Had Errors: NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 15

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Joins per Type:
Full table scans (Select_scan): 1
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0

Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0

Index Usage:
No Index used

Other Info:
Event Id: 2276
Thread Id: 69

CREATE INDEX oppurtunity_index ON opportunity (opp_title);
SELECT * from opportunity where opp_title = 'Software Development Internship';

Timing (as measured at client side):
Execution time: 0:00:0.00000000

Timing (as measured by the server):
Execution time: 0:00:0.00056990
Table lock wait time: 0:00:0.00000400

Errors:
Had Errors: NO
Warnings: 0

Rows Processed:
Rows affected: 0
Rows sent to client: 1
Rows examined: 1

Temporary Tables:
Temporary disk tables created: 0
Temporary tables created: 0

Joins per Type:
Full table scans (Select_scan): 0
Joins using table scans (Select_full_join): 0
Joins using range search (Select_full_range_join): 0
Joins with range checks (Select_range_check): 0
Joins using range (Select_range): 0

Sorting:
Sorted rows (Sort_rows): 0
Sort merge passes (Sort_merge_passes): 0
Sorts with ranges (Sort_range): 0
Sorts with table scans (Sort_scan): 0

Index Usage:
At least one Index was used

Other Info:
Event Id: 2402
Thread Id: 69

MYSQL INDEX QUERIES

1. CREATE INDEX student_index ON student (student_first_name,student_middle_name,student_last_name);
2. CREATE INDEX placement_index ON placement (company_name);
3. CREATE INDEX intership_index ON internship (company_name);
4. CREATE INDEX company_index ON company (company_name);
5. CREATE INDEX opportunity_index ON opportunity (opp_title);
6. CREATE INDEX point_of_contact_index ON point_of_contact(opp_id);
7. CREATE INDEX selection_procedure_index ON selection_procedure(round_number,round_type);
8. CREATE INDEX requirements_index ON requirements(min_cpi_req,opp_status,program_req, dept_req);
9. CREATE INDEX application_index ON application(student_id);
10. CREATE INDEX app_opp_index ON app_opp(student_id,status);

User Defined Data Type:

MySQL implementation does not support user defined data types, however we have used ENUM to specify a few of the attributes which can take only a few of the values.

1. **opp_type ENUM('internship', 'placement') NOT NULL**: (in table 'opportunity')
Opportunity can be an internship or placement, so this data type creates a column named opp_type with only two possible values, internship or placement. The ENUM data type restricts the column to accept only the specified values. The NOT NULL constraint ensures that a value must be provided for this column.

2. **min_cpi_req DECIMAL(3, 1) CHECK (min_cpi_req >= 0 AND min_cpi_req <= 10)** (in table 'requirements')
This field stores a minimum CPI requirement for a certain opportunity. If the 'min_cpi_req' field is included in the 'requirements' table with the above data type and constraint, it means that any opportunity that has a minimum CPI requirement can be specified in this table and the minimum CPI requirement must be between 0 and 10.

3. **active_backlog ENUM('yes','no') NOT NULL**: (in table 'requirements')
Some companies specify requirements that students should not have active backlogs to sit for the opportunity, so active_backlog is set to 'yes'. If no such requirement is defined by the company then active_backlog is 'no'.

4. **program_req ENUM('btech','mtech','phd') NOT NULL**: (in table 'requirements')
This column can be used to indicate the educational qualification required for a job opportunity. For example, if the program_req is set to btech, it means that the candidate must be enrolled in a bachelor's degree in technology to be eligible for the job opportunity. Students can be enrolled in three possible programs - B.Tech, M.Tech, Ph.D. This data type creates a column named program_req with three possible values, btech, mtech, or phd.

5. **dept_req ENUM('CSE','EE','ME','CE','CVE','MSE', 'OTHER') NOT NULL**: (in table 'requirements')
The dept_req holds the different branches or departments, so dept_req column has with seven possible values, CSE, EE, ME, CE, CVE, MSE, or OTHER. 'OTHER' is used to specify any other possible departments as we are not aware of every department that the institute offers.

6. **year_req varchar(4) CHECK (year_req REGEXP '^[0-9]{4}$') NOT NULL** (in table 'requirements')
The data type year_req in the table requirements is defined as a VARCHAR (variable-length character string) with a length of 4. The CHECK constraint is used to ensure that the value of year_req matches a specific pattern. The pattern specified in this case is '^[0-9]{4}$', which is a regular expression. This regular expression specifies that year_req must be a string of exactly four digits, with no other characters allowed.

7. **opp_status ENUM('open', 'close') NOT NULL**: (in table ‘requirements’)
When an opportunity is created, it will typically be in the "open" state, which means that the opportunity is available for candidates to apply. Once the recruitment process is complete or the opportunity is no longer available, the status can be changed to "close" to indicate that the opportunity is no longer available. Using an ENUM data type for opp_status helps to ensure that the column only contains valid values, and it also makes it easier to write queries that filter on this column. For example, if you wanted to retrieve all the open opportunities, you could use a query like this:
```
**SELECT * FROM opportunities WHERE opp_status = 'open';**
```

8. **poc_email_id VARCHAR(255) CHECK (poc_email_id REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$') PRIMARY KEY** (in table ‘point_of_contact’)
This field stores the email address of the point of contact. CHECK is used to validate the email address format. The poc_email_id column is the primary key of the point_of_contact table. A primary key is a unique identifier for each row in the table, and it is used to enforce data integrity and ensure that there are no duplicate rows in the table. The PRIMARY KEY constraint also ensures that the poc_email_id column cannot contain NULL values.

9. **dept ENUM('CSE','EE','ME','CE','CVE','MSE') NOT NULL**: (in table ‘student’)
This data type creates a column named dept with six possible values, CSE, EE, ME, CE, CVE, or MSE.

10. **CPI DECIMAL(3,2) CHECK (CPI >= 0 AND CPI <= 10) NOT NULL**: (in table ‘student’)
CPI stands for Cumulative Performance Index, and it is a numeric value that represents the academic performance of a student.
DECIMAL(3,2) means that the CPI column will have a maximum of 3 digits, with 2 of them being decimal places. So, the CPI value can range from 0.00 to 10.00.
CHECK (CPI >= 0 AND CPI <= 10) is a constraint that ensures that only values between 0.00 and 10.00 are inserted into the CPI column. If a value outside this range is inserted, an error will be thrown.
NOT NULL means that the CPI column cannot be left empty or null. A valid CPI value must be provided for each row in the table.

11. **active_backlogs ENUM('yes','no') NOT NULL**: (in table ‘student’)
Backlogs refer to the subjects that a student has failed to clear a course in a particular semester or academic year. Backlog for student is binary and active_backlog can be used to check whether a candidate has any pending or active backlogs at the time of applying for an opportunity. A student can have active backlogs or not so, this data type creates a column named active_backlog with two possible values, yes or no.

12. **gender ENUM('male','female','other') NOT NULL**: (in table ‘student’)
This is a field where the gender of the student can be stored. Any record that is inserted into the "student" table must include a value for the "gender" column, which must be one of the specified options - 'male', 'female', or 'other'.

13. **method ENUM('ppo','interview') NOT NULL**: (in table ‘placement’)
Student can be placed through two methods - PPO (pre-placement offer) or interview. If the student is placed through PPO they cannot sit in interview process. This data type creates a column named method with two possible values, ppo or interview.

14. **medium_ ENUM('internal', 'external') NOT NULL**: (in table ‘placement’)
This represents the medium through which the opportunity was made available to the candidate. It is an enumerated data type with two possible values - 'internal' and 'external'.
If the value of medium_ is 'internal', it suggests that the opportunity was communicated to the candidate through internal channels such as the company's website, employee referrals, or the company's career portal. On the other hand, if the value of

medium_ is 'external', it implies that the opportunity was communicated to the candidate through external channels such as job portals, social media, or third-party recruiters.

Note: Here, we have used “medium_” instead of “medium” because “medium” is a keyword in MySQL language.

15. **method ENUM('internal’, ‘external’) NOT NULL (in table ‘internship’)**
This column represents the method of recruitment or selection for the internship opportunity. This is similar to the ‘method’ in table placement.

16. **status ENUM('eligible','not eligible','selected','rejected') NOT NULL (in table ‘app_opp’)**
This column is used to track the progress of an applicant's application for an opportunity and whether they are eligible for the opportunity or not. If an applicant is selected, then the application status will be updated accordingly, and if they are rejected, the status will be set to 'rejected'.

17. **resume_file VARCHAR(255) CHECK (resume_file REGEXP '^(http|https)://.+') NOT NULL (in table ‘resume’)**
This field is for the resume file of a student storing a string of up to 255 characters. The regular expression pattern specified in this case is '^(http|https)://.+'. This pattern is used to validate that the value of resume_file is a valid URL starting with either "http" or "https". The ^ symbol matches the beginning of the string, (http|https) matches either "http" or "https", :// matches the colon and two forward slashes that are part of a URL, and .+ matches one or more characters after the protocol and domain name.

Table Extension:

```
CREATE TABLE CSE as (select * from student where dept = 'CSE');
CREATE TABLE EE as (select * from student where dept = 'EE');
CREATE TABLE ME as (select * from student where dept = 'ME');
CREATE TABLE CE as (select * from student where dept = 'CE');
CREATE TABLE CVE as (select * from student where dept = 'CVE');
CREATE TABLE MSE as (select * from student where dept = 'MSE');
```

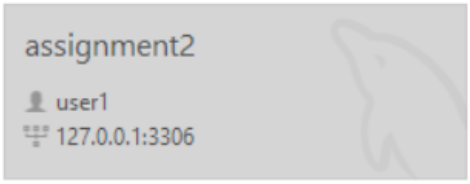
These are the queries to make new tables with the same schema. The main purpose of doing this table extension is to optimize the database's performance. The table extension allows for better organization and management of data. For example, we can easily retrieve a list of all students in a particular department, or we can obtain the number of students enrolled in each department. This type of data analysis can be beneficial in making decisions related to hiring, and stats.

Overall, the table extension of the student using department is optimizing the database's performance, ensuring data consistency, and providing efficient data organization and management.

3.2 Responsibility of G2:

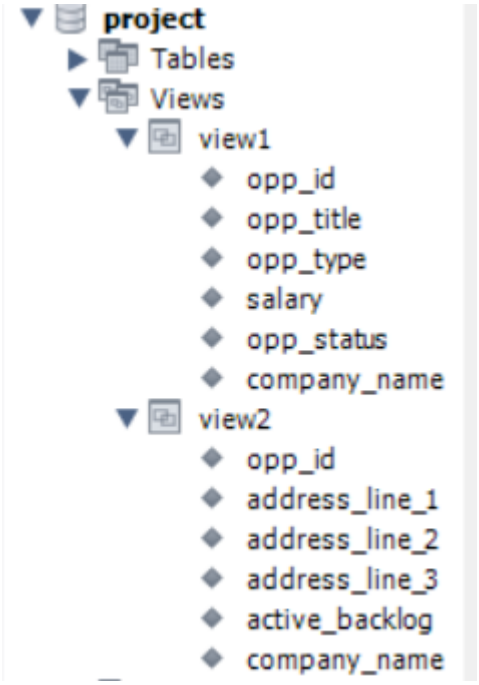
1. Create a user named "user1" with the password "password1".
We use the following sql command to create new user with given password:

CREATE USER 'user1'@'localhost' IDENTIFIED BY 'password1';



User with name “user1” and password “password1” created

2. Create Views on any of the two tables formed by G1 as view1 and view2. And make sure that views contain columns from at least two tables and one additional column with the user-defined data type.



We have used the tables opportunity and requirements table. Both the views

Opp_type and active_backlog are user-defined attributes. We have used three tables to create the views. These tables are:

1. Company
2. Opportunity
3. Requirements

View1 contains the following columns:

1. opp_id
2. opp_title
3. opp_type
4. salary
5. opp_status
6. company_name

View 2 contains the following columns:

1. opp_id
2. address_line_1
3. address_line_2
4. address_line_3
5. active_backlog
6. company_name

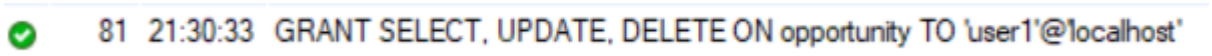
Note: Create type command is not supported by MySQL, therefore, we used an enum for creating user-defined data.

3. Grant "user1" the following permissions on "table1":

- **SELECT**
- **UPDATE**
- **DELETE**

We grant the permission using the root:

```
-- Q3
GRANT SELECT, UPDATE, DELETE ON opportunity TO 'user1'@'localhost';
```



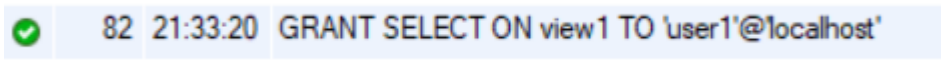
User has been granted permission on “table1”(Opportunity) to select, delete and update

4. Grant "user1" the following permissions on "view1":

- **SELECT**

We grant the permission using the root:

```
-- Q4
GRANT SELECT ON view1 TO 'user1'@'localhost';
```



User has been granted permission on “view1” to select.

5. Try to perform SELECT, UPDATE, and DELETE operations on "table1" and "view1" as "user1" and report your findings.

SELECT:

SELECT * FROM opportunity;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM view1;

opp_id	opp_title	opp_type	salary	opp_status	company_name
1	Software Engineering Intern	internship	500000	open	Apple Inc.
2	Marketing Manager	placement	700000	open	Microsoft Corporation
3	Data Science Intern	internship	800000	open	Alphabet Inc.
4	Sales Representative	placement	600000	open	Tesla Inc.
5	Product Management Intern	internship	550000	open	Amazon.com Inc.
6	Software Engineering Internship	internship	650000	open	Samsung Electronics
7	Hardware Engineer Placement	placement	550000	open	Toyota Motor Corporation
8	Data Science Internship	internship	600000	open	The Coca-Cola Company
9	Marketing Placement	placement	850000	open	Nestle S.A.
10	Software Development Internship	internship	700000	open	Apple Inc.
11	Product Manager Placement	placement	600000	open	Microsoft Corporation
12	Data Analyst Internship	internship	500000	open	Microsoft Corporation
13	Hardware Design Placement	placement	800000	open	Alphabet Inc.
14	Mobile App Development Internship	internship	650000	open	Tesla Inc.
15	Business Development Placement	placement	550000	open	Amazon.com Inc.

In Q3 and Q4, we have granted access to SELECT from “table1”(opportunity table) and “view1” to user1 thus, we are able to run the queries for both the tables.

UPDATE

UPDATE opportunity SET opp_title = 'New Intern' WHERE opp_id = 1;

```
4  22:13:29  UPDATE opportunity SET opp_title = 'New Intern' WHERE opp_id = 1   1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0
```

Here we can see that it has been updated
Before:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5

After:

	opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
▶	1	internship	New Intern	123 Main St.			1
	2	placement	Marketing Manager	456 Oak Ave.			2
	3	internship	Data Science Intern	789 Elm St.			3
	4	placement	Sales Representative	246 Maple Rd.			4
	5	internship	Product Management Intern	135 Cedar Ln.			5

UPDATE view1 SET company_name = 'New Company Name' WHERE opp_id = 1;

5 22:41:29 UPDATE view1 SET company_name = 'New Company Name' WHERE opp_id = 1 Error Code: 1142. UPDATE command denied to user 'user1'@'localhost' for table 'view1'

Here we can see that it has not been updated

	opp_id	opp_title	opp_type	salary	opp_status	company_name
▶	1	New Intern	internship	500000	open	Apple Inc.
	2	Marketing Manager	placement	700000	open	Microsoft Corporation
	3	Data Science Intern	internship	800000	open	Alphabet Inc.
	4	Sales Representative	placement	600000	open	Tesla Inc.
	5	Product Management Intern	internship	550000	open	Amazon.com Inc.
	6	Software Engineering Internship	internship	650000	open	Samsung Electronics
	7	Hardware Engineer Placement	placement	550000	open	Toyota Motor Corporation
	8	Data Science Internship	internship	600000	open	The Coca-Cola Company
	9	Marketing Placement	placement	850000	open	Nestle S.A.
	10	Software Development Internship	internship	700000	open	Apple Inc.
	11	Product Manager Placement	placement	600000	open	Microsoft Corporation
	12	Data Analyst Internship	internship	500000	open	Microsoft Corporation
	13	Hardware Design Placement	placement	800000	open	Alphabet Inc.
	14	Mobile App Development Internship	internship	650000	open	Tesla Inc.
	15	Business Development Placement	placement	550000	open	Amazon.com Inc.

Here, we can see that the query ran for “table1”(opportunity table) as in Q3. We granted user1 to update in it, while for view1 we have only granted the permission to select. Therefore, the update was made in opportunity table as we can see but not in “view1” because the access to UPDATE in “view1” is not granted to “user1”. Thus, while querying update for view1 it throws an error.

An important aspect to note is that the update made in “table1”(opportunity table” is reflected in “view1” too.

DELETE

DELETE FROM opportunity WHERE opp_id = 1;

7 22:46:17 DELETE FROM opportunity WHERE opp_id = 1 1 row(s) affected

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

DELETE FROM view1 WHERE opp_id = 2;

1 23:45:32 DELETE FROM view1 WHERE opp_id = 2 Error Code: 1142. DELETE command denied to user 'user1'@'localhost' for table 'view1'

opp_id	opp_title	opp_type	salary	opp_status	company_name
2	Marketing Manager	placement	700000	open	Microsoft Corporation
3	Data Science Intern	internship	800000	open	Alphabet Inc.
4	Sales Representative	placement	600000	open	Tesla Inc.
5	Product Management Intern	internship	550000	open	Amazon.com Inc.
6	Software Engineering Internship	internship	650000	open	Samsung Electronics
7	Hardware Engineer Placement	placement	550000	open	Toyota Motor Corporation
8	Data Science Internship	internship	600000	open	The Coca-Cola Company
9	Marketing Placement	placement	850000	open	Nestle S.A.
10	Software Development Internship	internship	700000	open	Apple Inc.
11	Product Manager Placement	placement	600000	open	Microsoft Corporation
12	Data Analyst Internship	internship	500000	open	Microsoft Corporation
13	Hardware Design Placement	placement	800000	open	Alphabet Inc.

As in “table1”(opportunity table) user1 is granted with the permission to DELETE thus, the first query gets run smoothly and entries where opp_id=1 gets deleted but as we havent granted DELETE access for “view1” to user1 thus, it throws an error and the query to delete opp_id=2 doesnt run and the table looks the same as previous.

6. Revoke the UPDATE and DELETE permissions on "table1" for "user1" and report your findings.

1 23:41:58 REVOKE UPDATE, DELETE ON opportunity FROM 'user1'@'localhost' 0 row(s) affected

We get to see that user 1 has lost the permission to update or delete from the opportunity table. We perform the following operations on the opportunity table with user1 as the user.

On performing update operation:

2 23:08:32 UPDATE opportunity SET opp_title = 'Again New Opportunity' WHERE opp_id = 2 Error Code: 1142. UPDATE command denied to user 'user1'@'localhost' for table 'opportunity'

On performing delete operation:

3 23:11:27 DELETE FROM opportunity WHERE opp_id = 2 Error Code: 1142. DELETE command denied to user 'user1'@'localhost' for table 'opportunity'

On performing select operation:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

As we can see that it throws an error in case of UPDATE and DELETE, this is because the access to update and delete from the “table1”(opportunity table) has been revoked for “user1”. Here opp_id=1 row is deleted as we had given that query in Q5.

7. Try to perform SELECT, UPDATE, and DELETE operations on "table1" and "view1" as "user1" again.

SELECT
For “table1”

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Note: Row with opp_id=1 is not there because we had deleted it in Q5.

For “view1”

opp_id	opp_title	opp_type	salary	opp_status	company_name
2	Marketing Manager	placement	700000	open	Microsoft Corporation
3	Data Science Intern	internship	800000	open	Alphabet Inc.
4	Sales Representative	placement	600000	open	Tesla Inc.
5	Product Management Intern	internship	550000	open	Amazon.com Inc.
6	Software Engineering Internship	internship	650000	open	Samsung Electronics
7	Hardware Engineer Placement	placement	550000	open	Toyota Motor Corporation
8	Data Science Internship	internship	600000	open	The Coca-Cola Company
9	Marketing Placement	placement	850000	open	Nestle S.A.
10	Software Development Internship	internship	700000	open	Apple Inc.
11	Product Manager Placement	placement	600000	open	Microsoft Corporation
12	Data Analyst Internship	internship	500000	open	Microsoft Corporation
13	Hardware Design Placement	placement	800000	open	Alphabet Inc.
14	Mobile App Development Internship	internship	650000	open	Tesla Inc.
15	Business Development Placement	placement	550000	open	Amazon.com Inc.

Here too row corresponding to opp_id=1 is dropped as “view1” contains elements from “table1” from where we deleted that row; thus its reflected here.
As the access to select has been granted for both the tables thus the query to SELECT is run for both “table1”(opportunity table) and “view1”.

UPDATE

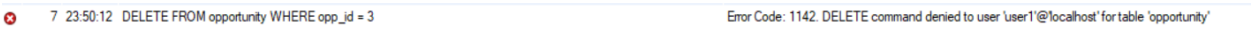
For “table1”

For “view1”

For table1 the user was granted permission to update in Q3 but this access was revoked in Q6 thus, now user1 will be unable to UPDATE form opportunity table in table1 and an error will be shown for the same. For “view1” user1 never had the access to update in it thus it shows an error similar to above questions.

DELETE

For “table1”



For “view1”

For table1, the user was granted permission to delete in Q3, but this access was revoked in Q6 thus, now user1 will be unable to DELETE form opportunity table in “table1,” and error will be shown for the same. For “view1,” user1 never had access to update in it thus it shows an error similar to above questions.

2. Mention the situation that violates the referential integrity, show the updates in the table, and how we can solve such problems.
Make sure to document all the steps and the results, along with screenshots of each step, clearly and comprehensively. Document the steps taken to complete this assignment and include screenshots as evidence of the permissions granted and revoked.

1. opportunity and requirements tables
We have the 'opp_id' as the primary key of the opportunity table, which acts as a foreign key under the attribute 'opp_id' of the requirements table, i.e., requirements table is referencing the opportunity table.

So, if we delete a record from the opportunity table, then the corresponding record from the requirements table where the foreign key constraint's value is equal to the primary key value of the deleted record should also get deleted.

The following table is the opportunity table populated by 15 records initially.

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5

The following table is the requirements table populated by 15 records initially.

req_id	opp_id	min_cpi_req	active_backlog	program_req	dept_req	year_req	salary	opp_status	opp_req	posted_on	deadline
1	1	8.0	yes	btech	CSE	2023	500000	open	Strong programming skills in Java required	2023-03-01	2023-03-31
2	2	7.5	no	mtech	EE	2022	700000	open	Experience with Google Analytics required	2023-02-15	2023-03-15
3	3	8.5	no	phd	ME	2024	800000	open	Experience with Python and data analysis requi...	2023-03-10	2023-04-30
4	4	7.0	yes	btech	CVE	2023	600000	open	Experience in sales or marketing preferred	2023-03-05	2023-04-05
5	5	8.0	no	btech	MSE	2023	550000	open	Strong analytical and problem-solving skills requi...	2023-03-08	2023-04-08
6	6	7.5	yes	mtech	CSE	2022	650000	open	Experience with machine learning algorithms an...	2023-03-12	2023-04-12
7	7	8.0	no	btech	EE	2023	550000	open	Experience with embedded systems programmin...	2023-03-14	2023-04-14
8	8	7.0	yes	btech	ME	2023	600000	open	Strong communication skills and ability to work i...	2023-03-16	2023-04-16
9	9	8.5	no	phd	CSE	2024	850000	open	Experience with natural language processing an...	2023-03-18	2023-04-18
10	10	7.5	no	mtech	CVE	2022	700000	open	Experience with project management and agile ...	2023-03-20	2023-04-20
11	11	8.0	yes	btech	MSE	2023	600000	open	Strong understanding of data structures and al...	2023-03-22	2023-04-22
12	12	7.0	no	btech	EE	2023	500000	open	Experience with PCB design and testing required	2023-03-24	2023-04-24
13	13	8.5	no	phd	ME	2024	800000	open	Experience with finite element analysis and simu...	2023-03-26	2023-04-26
14	14	7.5	yes	mtech	EE	2022	650000	open	Experience with RF circuit design and testing re...	2023-03-28	2023-04-28
15	15	8.0	no	btech	OTHER	2023	550000	open	Strong understanding of web development and ...	2023-03-30	2023-04-30

If we delete a record from the opportunity table with opp_id = 3, then we get the following result.

COMMAND: DELETE FROM opportunity WHERE opp_id = 3;

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5

Since there is a reference relation between requirements and opportunity table, the record in the requirements table where the foreign key constraint is opp_id = 3 should get deleted. So after performing the above delete operation, if we see the requirements table, it is as follows:

req_id	opp_id	min_cpi_req	active_backlog	program_req	dept_req	year_req	salary	opp_status	opp_req	posted_on	deadline
1	1	8.0	yes	btech	CSE	2023	500000	open	Strong programming skills in Java required	2023-03-01	2023-03-31
2	2	7.5	no	mtech	EE	2022	700000	open	Experience with Google Analytics required	2023-02-15	2023-03-15
3	3	8.5	no	phd	ME	2024	800000	open	Experience with Python and data analysis requi...	2023-03-10	2023-04-30
4	4	7.0	yes	btech	CVE	2023	600000	open	Experience in sales or marketing preferred	2023-03-05	2023-04-05
5	5	8.0	no	btech	MSE	2023	550000	open	Strong analytical and problem-solving skills requi...	2023-03-08	2023-04-08
6	6	7.5	yes	mtech	CSE	2022	650000	open	Experience with machine learning algorithms an...	2023-03-12	2023-04-12
7	7	8.0	no	btech	EE	2023	550000	open	Experience with embedded systems programmin...	2023-03-14	2023-04-14
8	8	7.0	yes	btech	ME	2023	600000	open	Strong communication skills and ability to work i...	2023-03-16	2023-04-16
9	9	8.5	no	phd	CSE	2024	850000	open	Experience with natural language processing an...	2023-03-18	2023-04-18
10	10	7.5	no	mtech	CVE	2022	700000	open	Experience with project management and agile ...	2023-03-20	2023-04-20
11	11	8.0	yes	btech	MSE	2023	600000	open	Strong understanding of data structures and al...	2023-03-22	2023-04-22
12	12	7.0	no	btech	EE	2023	500000	open	Experience with PCB design and testing required	2023-03-24	2023-04-24
13	13	8.5	no	phd	ME	2024	800000	open	Experience with finite element analysis and simu...	2023-03-26	2023-04-26
14	14	7.5	yes	mtech	EE	2022	650000	open	Experience with RF circuit design and testing re...	2023-03-28	2023-04-28
15	15	8.0	no	btech	OTHER	2023	550000	open	Strong understanding of web development and ...	2023-03-30	2023-04-30

The record which had opp_id = 3 is still there in the table, it didn't get deleted. So, this is a violation of referential integrity. In order to solve this problem, we will add the 'ON DELETE CASCADE' option in the foreign key constraint definition between the two tables.

Adding: foreign key (opp_id) references opportunity(opp_id) on delete cascade while defining the requirements table schema.

Now, if we run the delete command and check the requirements table, we get the following result.

req_id	opp_id	min_cpi_req	active_backlog	program_req	dept_req	year_req	salary	opp_status	opp_req	posted_on	deadline
1	1	8.0	yes	btech	CSE	2023	500000	open	Strong programming skills in Java required	2023-03-01	2023-03-31
2	2	7.5	no	mtech	EE	2022	700000	open	Experience with Google Analytics required	2023-02-15	2023-03-15
4	4	7.0	yes	btech	CVE	2023	600000	open	Experience in sales or marketing preferred	2023-03-05	2023-04-05
5	5	8.0	no	btech	MSE	2023	550000	open	Strong analytical and problem-solving skills requi...	2023-03-08	2023-04-08
6	6	7.5	yes	mtech	CSE	2022	650000	open	Experience with machine learning algorithms an...	2023-03-12	2023-04-12
7	7	8.0	no	btech	EE	2023	550000	open	Experience with embedded systems programmin...	2023-03-14	2023-04-14
8	8	7.0	yes	btech	ME	2023	600000	open	Strong communication skills and ability to work i...	2023-03-16	2023-04-16
9	9	8.5	no	phd	CSE	2024	850000	open	Experience with natural language processing an...	2023-03-18	2023-04-18
10	10	7.5	no	mtech	CVE	2022	700000	open	Experience with project management and agile ...	2023-03-20	2023-04-20
11	11	8.0	yes	btech	MSE	2023	600000	open	Strong understanding of data structures and al...	2023-03-22	2023-04-22
12	12	7.0	no	btech	EE	2023	500000	open	Experience with PCB design and testing required	2023-03-24	2023-04-24
13	13	8.5	no	phd	ME	2024	800000	open	Experience with finite element analysis and simu...	2023-03-26	2023-04-26
14	14	7.5	yes	mtech	EE	2022	650000	open	Experience with RF circuit design and testing re...	2023-03-28	2023-04-28
15	15	8.0	no	btech	OTHER	2023	550000	open	Strong understanding of web development and ...	2023-03-30	2023-04-30

This time, the record where opp_id = 3 got deleted from the requirements table, hence solving the problem of violation of the referential integrity.

2. opportunity and company tables

We have the ‘company_id’ as the primary key of the company table, which acts as a foreign key under the attribute ‘company_id’ of the opportunity table, i.e., opportunity table is referencing the company table.

So, if we delete a record from the company table, then the corresponding record from the opportunity table where the foreign key constraint’s value is equal to the primary key value of the deleted record should also get deleted.

Initial state of company table:

company_id	company_name	company_website	organization_type	industry_type	country	company_pincode	address_line_1	address_line
1	Apple Inc.	https://www.apple.com/	Corporation	Technology	USA	95014	1 Infinite Loop	Cupertino
2	Microsoft Corporation	https://www.microsoft.com/	Corporation	Technology	USA	98052	One Microsoft Way	Redmond
3	Alphabet Inc.	https://abc.xyz/	Corporation	Technology	USA	94043	1600 Amphitheatre Parkway	Mountain Vie
4	Tesla Inc.	https://www.tesla.com/	Corporation	Automotive	USA	94304	3500 Deer Creek Road	Palo Alto
5	Amazon.com Inc.	https://www.amazon.com/	Corporation	Retail	USA	98109	410 Terry Avenue North	Seattle
6	Samsung Electronics	https://www.samsung.com/	Corporation	Technology	South Korea	135-090	129, Samsung-ro	Yeongtong-g
7	Toyota Motor Corporation	https://global.toyota/en/	Corporation	Automotive	Japan	471-8571	1 Toyota-cho	Toyota City
8	The Coca-Cola Company	https://www.coca-colacompany.com/	Corporation	Beverage	USA	30313	One Coca-Cola Plaza	Atlanta
9	Nestle S.A.	https://www.nestle.com/	Corporation	Food	Switzerland	1800	Avenue Nestle 55	Vevey
10	Boeing Company	https://www.boeing.com/	Corporation	Aerospace	USA	60616	100 North Riverside	Chicago
11	Procter & Gamble Co.	https://us.pg.com/	Corporation	Consumer go...	USA	45202	One Procter & Gamble Plaza	Cincinnati
12	Ford Motor Company	https://www.ford.com/	Corporation	Automotive	USA	48126	20900 Oakwood Boulevard	Dearborn
13	Intel Corporation	https://www.intel.com/	Corporation	Technology	USA	95052	2200 Mission College Boule...	Santa Clara
14	McDonalds Corporation	https://www.mcdonalds.com/	Corporation	Fast food	USA	60606	110 North Carpenter Street	Chicago
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Initial state of opportunity table:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

If we delete a record from the company table with company_id = 1, then we get the following result.

COMMAND: DELETE FROM company where company_id = 1;

company_id	company_name	company_website	organization_type	industry_type	country	company_pincode	address_line_1	address_
2	Microsoft Corporation	https://www.microsoft.com/	Corporation	Technology	USA	98052	One Microsoft Way	Redmond
3	Alphabet Inc.	https://abc.xyz/	Corporation	Technology	USA	94043	1600 Amphitheatre Parkway	Mountain
4	Tesla Inc.	https://www.tesla.com/	Corporation	Automotive	USA	94304	3500 Deer Creek Road	Palo Alto
5	Amazon.com Inc.	https://www.amazon.com/	Corporation	Retail	USA	98109	410 Terry Avenue North	Seattle
6	Samsung Electronics	https://www.samsung.com/	Corporation	Technology	South Korea	135-090	129, Samsung-ro	Yeongton
7	Toyota Motor Corporation	https://global.toyota/en/	Corporation	Automotive	Japan	471-8571	1 Toyota-cho	Toyota C
8	The Coca-Cola Company	https://www.coca-colacompany.com/	Corporation	Beverage	USA	30313	One Coca-Cola Plaza	Atlanta
9	Nestle S.A.	https://www.nestle.com/	Corporation	Food	Switzerland	1800	Avenue Nestle 55	Vevey
10	Boeing Company	https://www.boeing.com/	Corporation	Aerospace	USA	60616	100 North Riverside	Chicago
11	Procter & Gamble Co.	https://us.pg.com/	Corporation	Consumer goods	USA	45202	One Procter & Gamble Plaza	Cincinnat
12	Ford Motor Company	https://www.ford.com/	Corporation	Automotive	USA	48126	20900 Oakwood Boulevard	Dearborn
13	Intel Corporation	https://www.intel.com/	Corporation	Technology	USA	95052	2200 Mission College Boule...	Santa Cla
14	McDonalds Corporation	https://www.mcdonalds.com/	Corporation	Fast food	USA	60606	110 North Carpenter Street	Chicago
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Since there is a reference relation between opportunity and company table, the record in the placement table where the foreign key constraint is company_id = 1 should get deleted. So after performing the above delete operation, if we see the opportunity table, it is as follows:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The record which had company_id = 1 is still there in the table, it didn't got deleted. So, this is a violation of referential integrity.
In order to solve this problem, we will add the 'ON DELETE CASCADE' option in the foreign key constraint definition between the two tables.

Adding: foreign key (company_id) references company(company_id) on delete cascade while defining the opportunity table schema.

Now, if we run the delete command and check the opportunity table, we get the following result.

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_i
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Intern...	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5

3. student and placement tables
We have the 'student_id' as the primary key of the student table, which acts as a foreign key under the attribute 'student_id' of the placement table i.e., placement table is referencing the student table.

So, if we delete a record from the student table, then the corresponding record from the placement table where the foreign key constraint's value is equal to the primary key value of the deleted record should also get deleted.
Initial state of student table:

student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
1	Samantha		Johnson	NULL	CSE	9.80	no	female	3
2	Jack	Thomas	Doe	NULL	EE	7.50	yes	male	2
3	Sarah		Wilson	NULL	CE	8.20	no	female	4
4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2

Initial state of placement table:

placement_id	method	medium_	salary	company_name	student_id
1	ppo	internal	700000.00	Apple Inc.	1
2	interview	external	800000.00	XYZ Corp.	2

If we delete a record from the student table with student_id = 1, then we get the following result.
COMMAND: DELETE FROM student where student_id = 1;

student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
2	Jack	Thomas	Doe	NULL	EE	7.50	yes	male	2
3	Sarah		Wilson	NULL	CE	8.20	no	female	4
4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Since there is a reference relation between placement and student table, the record in the placement table where the foreign key constraint is student_id = 3 should get deleted. So after performing the above delete operation, if we see the placement table, it is as follows:

placement_id	method	medium_	salary	company_name	student_id
1	ppo	internal	700000.00	Apple Inc.	1
2	interview	external	800000.00	XYZ Corp.	2
NULL	NULL	NULL	NULL	NULL	NULL

The record which had student_id = 1 is still there in the table, it didn't got deleted. So, this is a violation of a referential integrity.
In order to solve this problem, we will add the 'ON DELETE CASCADE' option in the foreign key constraint definition between the two tables.

Adding: foreign key (student_id) references student(student_id) on delete cascade while defining the placement table schema.

Now, if we run the delete command and check the requirements table, we get the following result.

placement_id	method	medium_	salary	company_name	student_id
2	interview	external	800000.00	XYZ Corp.	2
NULL	NULL	NULL	NULL	NULL	NULL

4. opportunity and company tables
We have the 'company_id' as the primary key of the company table, which acts as a foreign key under the attribute 'company_id' of the opportunity table, i.e., opportunity table is referencing the company table.

So, if we update a record from the company table, then the corresponding record from the opportunity table where the foreign key constraint's value is equal to the primary key value of the updated record should also get updated.

Initial state of company table:

company_id	company_name	company_website	organization_type	industry_type	country	company_pincode	address_line_1	address_line
1	Apple Inc.	https://www.apple.com/	Corporation	Technology	USA	95014	1 Infinite Loop	Cupertino
2	Microsoft Corporation	https://www.microsoft.com/	Corporation	Technology	USA	98052	One Microsoft Way	Redmond
3	Alphabet Inc.	https://abc.xyz/	Corporation	Technology	USA	94043	1600 Amphitheatre Parkway	Mountain Vie
4	Tesla Inc.	https://www.tesla.com/	Corporation	Automotive	USA	94304	3500 Deer Creek Road	Palo Alto
5	Amazon.com Inc.	https://www.amazon.com/	Corporation	Retail	USA	98109	410 Terry Avenue North	Seattle
6	Samsung Electronics	https://www.samsung.com/	Corporation	Technology	South Korea	135-090	129, Samsung-ro	Yeongtong-g
7	Toyota Motor Corporation	https://global.toyota/en/	Corporation	Automotive	Japan	471-8571	1 Toyota-cho	Toyota City
8	The Coca-Cola Company	https://www.coca-colacompany.com/	Corporation	Beverage	USA	30313	One Coca-Cola Plaza	Atlanta
9	Nestle S.A.	https://www.nestle.com/	Corporation	Food	Switzerland	1800	Avenue Nestle 55	Vevey
10	Boeing Company	https://www.boeing.com/	Corporation	Aerospace	USA	60616	100 North Riverside	Chicago
11	Procter & Gamble Co.	https://us.pg.com/	Corporation	Consumer go...	USA	45202	One Procter & Gamble Plaza	Cincinnati
12	Ford Motor Company	https://www.ford.com/	Corporation	Automotive	USA	48126	20900 Oakwood Boulevard	Dearborn
13	Intel Corporation	https://www.intel.com/	Corporation	Technology	USA	95052	2200 Mission College Boule...	Santa Clara
14	McDonalds Corporation	https://www.mcdonalds.com/	Corporation	Fast food	USA	60606	110 North Carpenter Street	Chicago
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Initial state of opportunity table:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

If we delete a record from the company table with company_id = 1, then we get the following result.
COMMAND: **UPDATE company set company_id = 19 where company_id = 2;**

company_id	company_name	company_website	organization_type	industry_type	country	company_pincode	address_line_1	address_
1	Apple Inc.	https://www.apple.com/	Corporation	Technology	USA	95014	1 Infinite Loop	Cupertino
3	Alphabet Inc.	https://abc.xyz/	Corporation	Technology	USA	94043	1600 Amphitheatre Parkway	Mountain
4	Tesla Inc.	https://www.tesla.com/	Corporation	Automotive	USA	94304	3500 Deer Creek Road	Palo Alto
5	Amazon.com Inc.	https://www.amazon.com/	Corporation	Retail	USA	98109	410 Terry Avenue North	Seattle
6	Samsung Electronics	https://www.samsung.com/	Corporation	Technology	South Korea	135-090	129, Samsung-ro	Yeongtong
7	Toyota Motor Corporation	https://global.toyota/en/	Corporation	Automotive	Japan	471-8571	1 Toyota-cho	Toyota Cit
8	The Coca-Cola Company	https://www.coca-colacompany.com/	Corporation	Beverage	USA	30313	One Coca-Cola Plaza	Atlanta
9	Nestle S.A.	https://www.nestle.com/	Corporation	Food	Switzerland	1800	Avenue Nestle 55	Vevey
10	Boeing Company	https://www.boeing.com/	Corporation	Aerospace	USA	60616	100 North Riverside	Chicago
11	Procter & Gamble Co.	https://us.pg.com/	Corporation	Consumer goods	USA	45202	One Procter & Gamble Plaza	Cincinnati
12	Ford Motor Company	https://www.ford.com/	Corporation	Automotive	USA	48126	20900 Oakwood Boulevard	Dearborn
13	Intel Corporation	https://www.intel.com/	Corporation	Technology	USA	95052	2200 Mission College Boule...	Santa Clar
14	McDonalds Corporation	https://www.mcdonalds.com/	Corporation	Fast food	USA	60606	110 North Carpenter Street	Chicago
19	Microsoft Corporation	https://www.microsoft.com/	Corporation	Technology	USA	98052	One Microsoft Way	Redmond
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Since, there is a reference relation between opportunity and company table, the record in the placement table where the foreign key constraint is student_id = 1 should get deleted. So after performing the above delete operation, if we see the opportunity table, it is as follows:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The record which had company_id = 2 is still there in the table, it didn't got deleted. So, this is a violation of referential integrity.
In order to solve this problem, we will add the 'ON UPDATE CASCADE' option in the foreign key constraint definition between the two tables.

Adding: **foreign key (company_id) references company(company_id) on update cascade** while defining the opportunity table schema.

Now, if we run the update command and check the opportunity table, we get the following result.

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			19
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	19
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	19
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Opportunity and selection_procedure tables:
Primary key of Opportunity: opp_id
Primary key selection_procedure: (round_type, round_number)
selection_procedure: foreign key (opp_id) references opportunity(opp_id)

So, suppose we delete a record from the opportunity table. In that case, the corresponding record from the selection_procedure table where the foreign key constraint's value equals the primary key value of the deleted record should also get deleted.

Initial opportunity table:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5

Initial selection_procedure table:

round_type	round_venue_link	round_number	round_date	opp_id
Group Discussion	https://example.com/gd	1	2023-03-25	3
HR Interview	NULL	2	2023-03-30	4
Online Test	https://example.com/test	1	2023-03-15	1
Technical Interview	NULL	2	2023-03-20	2
Technical Test	https://example.com/tech-test	1	2023-04-05	5

If we delete a record from the opportunity table with opp_id = 3, then we get the following result.

COMMAND: **DELETE FROM opportunity WHERE opp_id = 3;**

round_type	round_venue_link	round_number	round_date	opp_id
Group Discussion	https://example.com/gd	1	2023-03-25	3
HR Interview	NULL	2	2023-03-30	4
Online Test	https://example.com/test	1	2023-03-15	1
Technical Interview	NULL	2	2023-03-20	2
Technical Test	https://example.com/tech-test	1	2023-04-05	5

Since there is a reference relation between selection_procedure and opportunity table, the record in the selection_procedure table where the foreign key constraint is opp_id = 3 should get deleted. The record which had opp_id = 3 is still there in the table; it didn't get deleted. So, this is a violation of referential integrity.
In order to solve this problem, we will add the 'ON DELETE CASCADE' option in the foreign key constraint definition between the two tables.

Adding:foreign key (opp_id) references opportunity(opp_id) on delete cascade while defining the requirements table schema.

Now, if we run the delete command and check the requirements table, we get the following result.

	round_type	round_venue_link	round_number	round_date	opp_id
▶	HR Interview	NULL	2	2023-03-30	4
	Online Test	https://example.com/test	1	2023-03-15	1
	Technical Interview	NULL	2	2023-03-20	2
	Technical Test	https://example.com/tech-test	1	2023-04-05	5
	NULL	NULL	NULL	NULL	NULL

This time, the record where opp_id = 3 got deleted from the selection_procedure table, hence solving the problem of violation of the referential integrity.

6. student and application tables

We have the 'student_id' as the primary key of the student table, which acts as a foreign key under the attribute 'student_id' of the application table, i.e., application table is referencing the student table.

So, if we delete a record from the student table, then the corresponding record from the application table where the foreign key constraint's value is equal to the primary key value of the deleted record, should also get deleted.

The following table is the student table populated by 5 records initially.

	student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
▶	1	Samantha		Johnson	NULL	CSE	9.80	no	female	3
	2	Jack	Thomas	Doe	NULL	EE	7.50	yes	male	2
	3	Sarah		Wilson	NULL	CE	8.20	no	female	4
	4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
	5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2

The following table is the application table populated by 2 records initially.

	student_id	opp_id	resume_id
▶	1	1	1
	2	2	2

If we delete a record from the student table with student_id = 2, then we get the following result.

COMMAND: DELETE FROM student WHERE student_id = 2;

	student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
▶	1	Samantha		Johnson	NULL	CSE	9.80	no	female	3
	3	Sarah		Wilson	NULL	CE	8.20	no	female	4
	4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
	5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2

Since, there is a reference relation between application and student table, the record in the application table where the foreign key constraint is student_id = 2 should get deleted. So after performing the above delete operation, if we see the application table, it is as follows:

	student_id	opp_id	resume_id
▶	1	1	1
	2	2	2

The record which had student_id = 2 is still there in the table, it didn't get deleted. So, this is a violation of referential integrity.

In order to solve this problem, we will add the ‘ON DELETE CASCADE’ option in the foreign key constraint definition between the two tables.

Adding: foreign key (student_id) references student (student_id) on delete cascade while defining the application table schema.
Now, if we run the delete command, and check the application table, we get the following result.

	student_id	opp_id	resume_id
▶	1	1	1

This time, the record where student_id = 2 got deleted from the application table, hence solving the problem of violation of the referential integrity.

7. student and internship tables
We have the ‘student_id’ as the primary key of the student table, which acts as a foreign key under the attribute ‘student_id’ of the internship table, i.e., internship table is referencing the student table.

So, if we update a record’s primary key from the student table, then the corresponding record from the internship table, where the foreign key constraint’s value is equal to the primary key value of the updated record, should also get updated.

The following table is the student table populated by 5 records initially.

	student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
▶	1	Samantha		Johnson	NULL	CSE	9.80	no	female	3
	2	Jack	Thomas	Doe	NULL	EE	7.50	yes	male	2
	3	Sarah		Wilson	NULL	CE	8.20	no	female	4
	4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
	5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2

The following table is the internship table populated by 2 records initially.

	internship_id	timeline	method	salary	company_name	student_id
▶	1	Summer 2022	internal	50000.00	Microsoft Corporation	3
	2	Fall 2022	external	45000.00	XYZ Corp	4

If we update the student_id to 7 from 3 in the student table, then we get the following result:

COMMAND: UPDATE student set student_id = 7 where student_id = 3;

	student_id	student_first_name	student_middle_name	student_last_name	student_image	dept	CPI	active_backlogs	gender	study_year
▶	1	Samantha		Johnson	NULL	CSE	9.80	no	female	3
	2	Jack	Thomas	Doe	NULL	EE	7.50	yes	male	2
	4	Jacob	Michael	Taylor	NULL	ME	6.90	yes	male	3
	5	Emily	Rose	Clark	NULL	MSE	9.10	no	female	2
	7	Sarah		Wilson	NULL	CE	8.20	no	female	4

Since there is a reference relation between internship and student table, the record in the internship table where the foreign key constraint is student_id = 3 should get updated to 7. So after performing the above update operation, if we see the internship table, it is as follows:

	internship_id	timeline	method	salary	company_name	student_id
▶	1	Summer 2022	internal	50000.00	Microsoft Corporation	3
	2	Fall 2022	external	45000.00	XYZ Corp	4

The record which had student_id = 3 is still there in the table; it didn't get updated. So, this is a violation of referential integrity.
In order to solve this problem, we will add the 'ON UPDATE CASCADE' option in the foreign key constraint definition between the two tables.

Adding: foreign key (student_id) references student (student_id) on update cascade while defining the internship table schema.

Now, if we run the update command and check the internship table, we get the following result.

internship_id	timeline	method	salary	company_name	student_id
1	Summer 2022	internal	50000.00	Microsoft Corporation	7
2	Fall 2022	external	45000.00	XYZ Corp	4

This time, the record where student_id = 3 got updated to 7 in the internship table, hence solving the problem of violation of the referential integrity.

8. Opportunity and point_of_contact tables:
Primary key of Opportunity: opp_id
Primary key point_of_contact: poc_email_id
point_of_contact: foreign key (opp_id) references opportunity(opp_id)

So, suppose we delete a record from the opportunity table. In that case, the corresponding record from the point_of_contact table where the foreign key constraint's value equals the primary key value of the deleted record should also get deleted.

Initial opportunity table:

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id
1	internship	Software Engineering Intern	123 Main St.			1
2	placement	Marketing Manager	456 Oak Ave.			2
3	internship	Data Science Intern	789 Elm St.			3
4	placement	Sales Representative	246 Maple Rd.			4
5	internship	Product Management Intern	135 Cedar Ln.			5
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5

Initial point_of_contact table:

employee_first_name	employee_middle_name	employee_last_name	employee_designation	opp_id	poc_email_id
David		Lee	Internship Coordinator	3	david.lee@example.com
Jane		Smith	Recruiter	2	jane.smith@example.com
John		Doe	HR Manager	1	john.doe@example.com
Michael	S	Brown	Placement Manager	5	michael.brown@example.com
Rachel	K	Johnson	Placement Coordinator	4	rachel.johnson@example.com
NULL	NULL	NULL	NULL	NULL	NULL

If we delete a record from the opportunity table with opp_id = 3, then we get the following result.

Command: **DELETE FROM opportunity WHERE opp_id = 3;**

employee_first_name	employee_middle_name	employee_last_name	employee_designation	opp_id	poc_email_id
David		Lee	Internship Coordinator	3	david.lee@example.com
Jane		Smith	Recruiter	2	jane.smith@example.com
John		Doe	HR Manager	1	john.doe@example.com
Michael	S	Brown	Placement Manager	5	michael.brown@example.com
Rachel	K	Johnson	Placement Coordinator	4	rachel.johnson@example.com

Since there is a reference relation between point_of_contact and opportunity table, the record in the point_of_contact table where the foreign key constraint is opp_id = 3 should get deleted. The record which had opp_id = 3 is still there in the table; it didn't get deleted. So, this is a violation of referential integrity.
In order to solve this problem, we will add the 'ON DELETE CASCADE' option in the foreign key constraint definition between the two tables.

Adding:foreign key (opp_id) references opportunity(opp_id) on delete cascade while defining the point_of_contact table schema.
Now, if we run the delete command and check the requirements table, we get the following result.

employee_first_name	employee_middle_name	employee_last_name	employee_designation	opp_id	poc_email_id
Jane		Smith	Recruiter	2	jane.smith@example.com
John		Doe	HR Manager	1	john.doe@example.com
Michael	S	Brown	Placement Manager	5	michael.brown@example.com
Rachel	K	Johnson	Placement Coordinator	4	rachel.johnson@example.com
NULL	NULL	NULL	NULL	NULL	NULL

This time, the record where opp_id = 3 got deleted from the point_of_contact table, hence solving the problem of violation of the referential integrity.

----------NOTE----------
SET sql_safe_updates=0; - This is to remove safe mode. Allows to delete any entry.

SET foreign_key_checks = 0; does not check foreign key constraints, we can add any data to tables with cyclic dependencies.
SET foreign_key_checks = 1; checks foreign key constrains.

Unique Key Constraint:
The primary key is a unique key which is used to uniquely identify the record. When we defined the primary key, and try to add tro records with the same primary key, then we will get an error message.

The following image shows error while trying to add two records with same primary key in student table:

```
INSERT INTO student (student_id, student_first_name, student_middle_name, student_last_name, student_image, dept, CPI, active_backlogs, gender, study_year)
VALUES
(1, 'Samantha', '', 'Johnson', LOAD_FILE('/Users/sanskarsharma/Desktop/sample_image/sample_image.png'), 'CSE', 9.8, 'no', 'female', 3),
(1, 'Samantha', '', 'Johnson', LOAD_FILE('/Users/sanskarsharma/Desktop/sample_image/sample_image.png'), 'CSE', 9.8, 'no', 'female', 3);
```

122 21:40:44 INSERT INTO student (student_id, student_first_name, student_middle_name, student_last_name, student_ima... Error Code: 1062. Duplicate entry '1' for key 'student.PRIMARY' 0.000 sec

3.3 Responsibility of G1 & G2:

Using the optimized ER diagram, write five operations, their corresponding nested SQL queries, and tuple relational calculus/relational algebra with the following specifications:

1. **Two queries must throw an error due to violation of constraints specified by G1.**
2. **One of the functions should involve storage of an image along with a caption into the database.**
3. **Include cases of natural join, outer join, renaming, two or more different kinds of aggregate functions, and case statements in one or more queries.**

Clearly mention the specifications each operation satisfies. For, e.g., if operation *f1* involves storing an image, write "*f1 satisfies specification 2*". Also, submit screenshots of the results.

Queries:

- Find the data of floated opportunities. We have been given opp_id.
  - **select * from opportunity left join point_of_contact;**
  - **opportunity ⋈ requirements**

- Find the maximum salary of a student placed.
  - **select max(salary) from placement';**
  - $\sigma_{max(salary)}(placement)$
- Find the total number of students placed in the year xyz.
  - **select count(student_id) from placement natural join student where study_year = xyz;**
- Find the resume of the student applied in opportunity opp_id abc and the student has student_id xyz.
  - **select resume_file from application natural join resume where student_id = 1 and opp_id = 1;**
- Find an average package of cse, ee, me etc students.
  - **select dept, avg(salary) from placement natural join student group by dept;**
- How many students qualified in Round 3 of opportunity opp_id abc.
  - **select count(student_id) from (selection_procedure natural join opportunity natural join application) where round_number>3;**

Query to add image:

INSERT INTO student (student_id, student_first_name, student_middle_name, student_last_name, student_image, dept, CPI, active_backlogs, gender, study_year)
VALUES
(1, 'Samantha', '', 'Johnson', LOAD_FILE('/Users/sanskarsharma/Desktop/sample_image/sample_image.png'), 'CSE', 9.8, 'no', 'female', 3);

Queries with errors due to violation of constraints:

1. We find the left outer join from opportunity to point_of_contact table.

Query: **SELECT * FROM opportunity LEFT OUTER JOIN point_of_contact ON opportunity.opp_id=point_of_contact.opp_id;**
At the time of querying not every opportunity has been assigned a poc.

Relational Algebra:

opportunity $\bowtie_{opportunity.opp_id = point_of_contact.opp_id}$point_of_contact

opp_id	opp_type	opp_title	address_line_1	address_line_2	address_line_3	company_id	employee_first_name	employee_middle_name	employee
1	internship	Software Engineering Intern	123 Main St.			1	John		Doe
2	placement	Marketing Manager	456 Oak Ave.			2	Jane		Smith
3	internship	Data Science Intern	789 Elm St.			3	David		Lee
4	placement	Sales Representative	246 Maple Rd.			4	Rachel	K	Johnson
5	internship	Product Management Intern	135 Cedar Ln.			5	Michael	S	Brown
6	internship	Software Engineering Internship	1 Infinite Loop	Cupertino	California	6	NULL	NULL	NULL
7	placement	Hardware Engineer Placement	One Microsoft Way	Redmond	Washington	7	NULL	NULL	NULL
8	internship	Data Science Internship	1600 Amphitheatre Parkway	Mountain View	California	8	NULL	NULL	NULL
9	placement	Marketing Placement	3500 Deer Creek Road	Palo Alto	California	9	NULL	NULL	NULL
10	internship	Software Development Internship	410 Terry Avenue North	Seattle	Washington	1	NULL	NULL	NULL
11	placement	Product Manager Placement	1 Infinite Loop	Cupertino	California	2	NULL	NULL	NULL
12	internship	Data Analyst Internship	One Microsoft Way	Redmond	Washington	2	NULL	NULL	NULL
13	placement	Hardware Design Placement	1600 Amphitheatre Parkway	Mountain View	California	3	NULL	NULL	NULL
14	internship	Mobile App Development Internship	3500 Deer Creek Road	Palo Alto	California	4	NULL	NULL	NULL
15	placement	Business Development Placement	410 Terry Avenue North	Seattle	Washington	5	NULL	NULL	NULL

Reasoning: Using the Left outer join operation

2. Operation: We are attempting to use the UNION operator to combine the results of two SELECT statements.The first SELECT statement selects columns from the student table, while the second SELECT statement selects columns from the placement table

SQL Query:

```
SELECT * FROM student WHERE dept = 'CSE'
UNION
SELECT * FROM placement WHERE method = 'ppo';
```

Relational Algebra:

$$\sigma_{dept="CSE"}(student) \cup \sigma_{method="ppo"}(placement)$$

The operation satisfies 1 (error due to violation of constraints specified by G1)

Error Code: 214 21:44:52 SELECT * FROM student WHERE dept = 'CSE' UNION SELECT * FROM placement WHERE method = 'ppo' Error Code: 1222. The used SELECT statements have a different number of columns

Reasoning: The two SELECT statements have different column names and data types, which will result in a type mismatch error when the UNION operator tries to combine the two result sets.
Specifically, the student table columns are student_id, student_first_name, student_last_name, dept, CPI, active_backlogs, gender, and study_year. The placement table columns are placement_id, method, medium_, salary, company_name, and student_id. The two tables have different column names, and the CPI and salary columns have different data types (DECIMAL(3,2) and DECIMAL(10,2), respectively)

3. To find the maximum salary from the placement table we will use RENAME, CARTESIAN PRODUCT and SET DIFFERENCE. The second half of the query selects set of all salaries less than maximum salary. Then we subtract the set from the total set of salaries.

Initial state of the placement table:

placement_id	method	medium_	salary	company_name	student_id
1	ppo	internal	700000.00	Apple Inc.	1
2	interview	external	800000.00	XYZ Corp.	2
NULL	NULL	NULL	NULL	NULL	NULL

SQL Query:

```
SELECT p.salary from placement p
WHERE p.salary NOT IN
(SELECT p2.salary from placement p2 CROSS JOIN (SELECT * from placement) d where p2.salary < d.salary)
```

Result of the query:

salary
800000.00

Relation Algebra:

$$\prod_{salary}(placement) - \prod_{placement.salary}(\sigma_{placement.salary<d.salary}(placement \times \rho_{d}(placement)))$$

Reasoning:
The second half of the query renames the original table and takes the cartesian product. After selecting the salaries from p2 which are less than d, we get a set which contains all salaries less than maximum salary. The set difference gives the maximum salary.

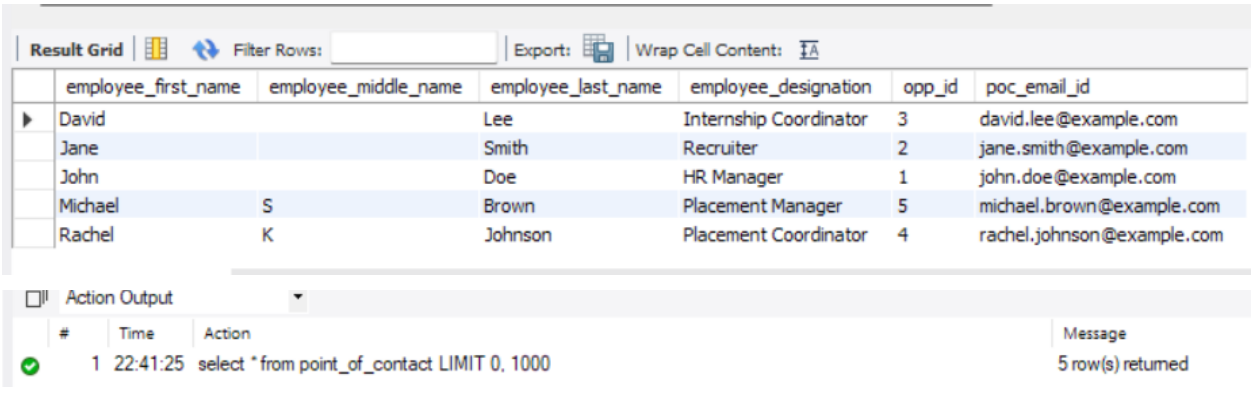
4. Operation: Inserting an entry into the point_of_contact table but having an invalid email id.

MySQL Query: **INSERT INTO point_of_contact VALUES ('Akash', '', 'Sharma', 'Manager', 10, 'akash.sharma.com')**

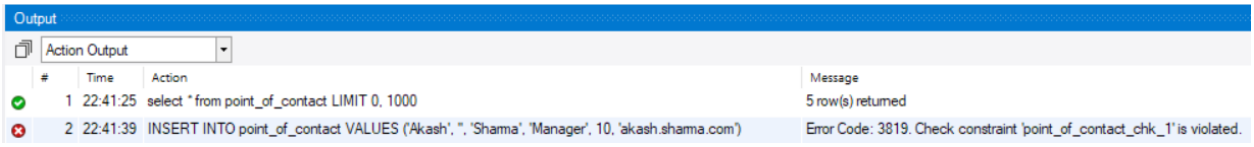
The operation satisfies 1 (error due to violation of constraints specified by G1)

Error Code: 3819. Check constraint 'point_of_contact_chk_1' is violated.

Before running query:



After running the query:



Reasoning:

We have set the constraints while defining the email attribute:
poc_email_id VARCHAR(255) CHECK (poc_email_id REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$') PRIMARY KEY

Here, the REGEXP defines that the constraint that email should be of the form : *something@something.something*. However, we are inserting 'akash.sharma.com'. Hence, it violates the constraint.

Here, the relational algebra does not exist for INSERT queries.

Commit Message:
We run the following query:
commit;

The following log message is shown:

LINK TO THE DRIVE HAVING RELEVANT FILES:
https://drive.google.com/drive/folders/1-_0TYhptemrrQ6yLUGt39VkRqG2te2mp?usp=sharing

---

Contributions:

G1:

Dhyeykumar Thummar (20110059)

- Wrote 1 SQL query which was invalid and violated the constraint specified in one of the attributes as a part of the combined work of G1 and G2.
- Assisted in writing and defining about all the user defined datatypes(*enum*).
- Helped in developing the queries and logic for image storage.
- Assisted in writing the relational algebra for few of the queries/operations.

Ksheer Sagar Agrawal (20110098)

- Designed indexing on all the schemas.
- Assisted in updating & improving relationship between schemas.
- Updated E-R Diagram.

Saatvik Rao (20110175)

- Wrote python script for generating random data for testing and analysis
- Populated all the tables with the random data
- Wrote and explained all the user-defined data types

Sanskar Sharma (20110185)

- Populated all tables with necessary data
- Improved database performance by normalizing the opportunity table
- Developed queries and logic for image storage in the database
- Rectified inconsistencies in the database to ensure data accuracy
- Helped in writing relational algebra queries

Mihir Sutariya (20110208)

- Created tables and applied constraints to ensure data integrity.
- Generated random student data for testing and analysis and executed a table extension query.
- These contributions were essential in creating a functional and efficient database.

Utkarsh Mishra (20110218)

- Helped in creating tables and constraints.
- Assisted with populating tables.
- Assisted in fixing inconsistencies.

G2:

Pushpendra Pratap Singh (20110151)

- Wrote 1 SQL query which shows the implementation of set difference, rename and cartesian product as a part of the combined work of G1 and G2.
- Picked up 2 pairs of tables (opportunity and company) and (student and placement) and showed foreign key constraint (violation of referential integrity) with screenshots.
- Updated constraints to prevent violation of refrential integrity.

R Yeeshu Dhurandhar (20110152)

- Picked up 2 pairs of tables (opportunity and selection_procedure) and (opportunity and point_of_contact) and showed foreign key constraint (violation of referential integrity) with screenshots.
- Combined the work of all three works and formatted the document.

Sahil Agrawal (20110178)

- Wrote 1 SQL query which shows error due to violation of constraints specified by G1.
- Picked up 2 pairs of tables (student and internship) and (opportunity and requirements) and showed foreign key constraint (violation of referential integrity) with screenshots.

Shruhrid Banthia (20110198)

- Wrote 1 SQL query which shows implementation of left outer join.
- Wrote queries to demonstrate the process of creating a user, and different views. Wrote queries to grant and revoke permissions to the user on tables and views. Documented the results.
- Rectified inconsistencies in the database to ensure data accuracy.

Tanvi Dixit (20110212)

- Wrote queries to demonstrate the process of creating a user,
- Granting and revoking different permissions on tables and views
- Helped with the documentation of the G2 report
- Searched for violation, errors and redundancy in G1 tables