

DBMS

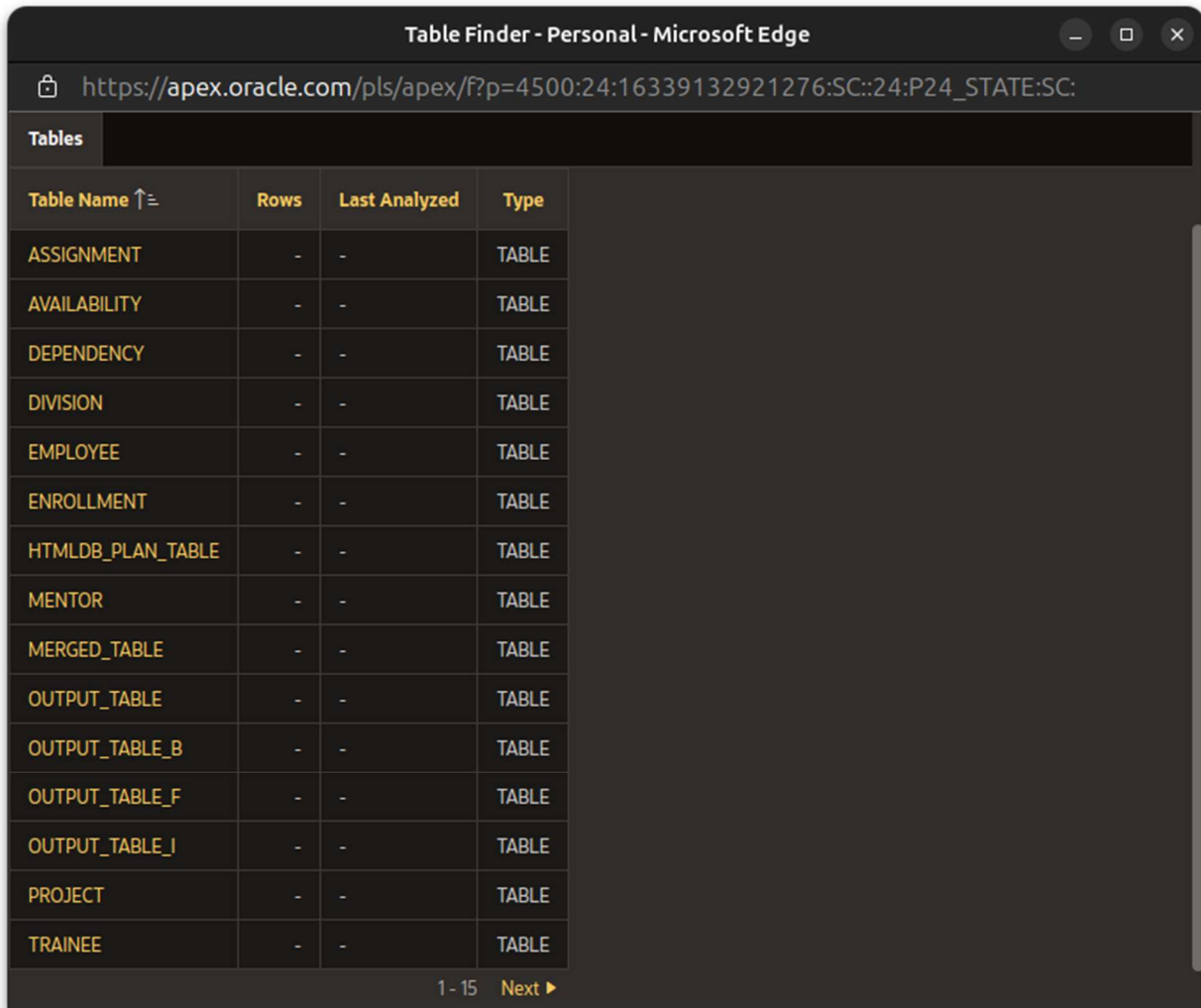
Assignment 6

Rhythm Baghel

B22CS042

University Projects Schema :

Output :



The screenshot shows the 'Table Finder - Personal - Microsoft Edge' window. The browser address bar displays the URL: https://apex.oracle.com/pls/apex/f?p=4500:24:16339132921276:SC::24:P24_STATE:SC:. The 'Tables' tab is selected, showing a table with the following columns: Table Name, Rows, Last Analyzed, and Type. The table lists 15 tables, all of which are of type 'TABLE'. The 'Rows' and 'Last Analyzed' columns contain dashes, indicating no data has been analyzed yet. The table is scrollable, and the bottom of the list shows '1 - 15' and a 'Next' button.

Table Name ↑	Rows	Last Analyzed	Type
ASSIGNMENT	-	-	TABLE
AVAILABILITY	-	-	TABLE
DEPENDENCY	-	-	TABLE
DIVISION	-	-	TABLE
EMPLOYEE	-	-	TABLE
ENROLLMENT	-	-	TABLE
HTMLDB_PLAN_TABLE	-	-	TABLE
MENTOR	-	-	TABLE
MERGED_TABLE	-	-	TABLE
OUTPUT_TABLE	-	-	TABLE
OUTPUT_TABLE_B	-	-	TABLE
OUTPUT_TABLE_F	-	-	TABLE
OUTPUT_TABLE_I	-	-	TABLE
PROJECT	-	-	TABLE
TRAINEE	-	-	TABLE

1 - 15 Next ▶

PL/SQL Queries :

- 1) Merge Tables: Write a PL/SQL block to merge the “PROJECT” and “DIVISION” tables with the “ASSIGNMENT” table to reduce redundancy.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a schema of 'WKSP_KARAN12345'. The language is set to 'PL/SQL' and the number of rows is set to '10'. The SQL command area contains the following PL/SQL block:

```
62 DECLARE
63     CURSOR c_merge IS
64         SELECT p.proj_id, p.proj_title, p.division_name, d.building_name, a.season, a.year, a.slot_id
65         FROM project p
66         JOIN division d ON p.division_name = d.division_name
67         JOIN assignment a ON p.proj_id = a.proj_id AND d.building_name = a.building_name;
68
69 BEGIN
70     FOR r IN c_merge LOOP
71         MERGE INTO merged_table mt
72         USING dual
73         ON (mt.proj_id = r.proj_id AND mt.building_name = r.building_name AND mt.slot_id = r.slot_id)
74         WHEN MATCHED THEN
75             UPDATE SET
76                 mt.proj_title = r.proj_title,
77                 mt.division_name = r.division_name,
78                 mt.season = r.season,
79                 mt.year = r.year
80         WHEN NOT MATCHED THEN
81             INSERT (proj_id, proj_title, division_name, building_name, season, year, slot_id)
82             VALUES (r.proj_id, r.proj_title, r.division_name, r.building_name, r.season, r.year, r.slot_id)
83     END LOOP;
84 END;
```

The 'Results' tab is selected, showing a table with 7 columns: PROJ_ID, PROJ_TITLE, DIVISION_NAME, BUILDING_NAME, SEASON, YEAR, and SLOT_ID. The table contains 7 rows of data:

PROJ_ID	PROJ_TITLE	DIVISION_NAME	BUILDING_NAME	SEASON	YEAR	SLOT_ID
1004	Quantum Circuit Design	Quantum Computing	Research Center	Winter	2023	104
1002	Robot Vision	Robotics	Tech Tower	Fall	2023	102
1001	Machine Learning Pipeline	Data Science	Tech Tower	Spring	2022	101
1004	Quantum Circuit Design	Quantum Computing	Research Center	Winter	2024	109
1001	Machine Learning Pipeline	Data Science	Tech Tower	Fall	2022	106
1002	Robot Vision	Robotics	Tech Tower	Summer	2023	107

The footer of the interface shows the user 'karan12345', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.

- 2) Output Employee Details: Write a PL/SQL block to output the emp_name, division_name, and proj_title for employees who work on projects with a duration longer than 12 months.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a script editor with a PL/SQL block. The script defines a cursor 'c_long_projects' and a loop to output employee details for projects with a duration greater than 12 months. The 'Results' tab at the bottom shows the output of the script, listing four employees and their associated project titles. The execution time is 0.03 seconds.

```
85
86 SELECT * FROM merged_table;
87
88 ----- 2 -----
89
90
91 DECLARE
92     CURSOR c_long_projects IS
93         SELECT e.emp_name, e.division_name, p.proj_title
94         FROM employee e
95         JOIN division d ON e.division_name = d.division_name
96         JOIN project p ON p.division_name = d.division_name
97         WHERE p.duration_months > 12; -- Assuming 'duration months' column in the project table
98
99
100 BEGIN
101     FOR r_employee IN c_long_projects LOOP
102         DBMS_OUTPUT.PUT_LINE('Employee Name: ' || r_employee.emp_name ||
103                               ', Division: ' || r_employee.division_name ||
104                               ', Project Title: ' || r_employee.proj_title);
105     END LOOP;
106 END;
107 /
```

Results Explain Describe Saved SQL History

Employee Name: Amit Verma, Division: AI Research, Project Title: AI Ethics Research
Employee Name: Vikram Sharma, Division: AI Research, Project Title: AI Ethics Research
Employee Name: Neha Singh, Division: AI Research, Project Title: AI Ethics Research
Employee Name: Prof. Suchetana Chakraborty, Division: AI Research, Project Title: AI Ethics Research

Statement processed.

0.03 seconds

karangeshtp@gmail.com karan12345 en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.1.3

- 3) Trainees with Mentor: Write a PL/SQL code to find the names of trainees (trainee_name) who have mentors from the same division.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a script editor with a PL/SQL code snippet. The code defines a cursor to find trainees and their mentors in the same division, then loops through the results to output the names. The 'Results' tab at the bottom shows the output of the script, listing trainee names, their divisions, and their mentors. The footer of the interface shows the user 'karanganeshtp@gmail.com', the schema 'karan12345', and the Oracle APEX version '24.1.3'.

```
105 END LOOP;  
106 END;  
107  
108  
109 ----- 3 -----  
110  
111 DECLARE  
112     CURSOR c_trainees_with_same_division IS  
113         SELECT t.trainee_name, t.division_name, e.emp_name AS mentor_name  
114         FROM trainee t  
115         JOIN mentor m ON t.trainee_ID = m.trainee_ID  
116         JOIN employee e ON m.mentor_ID = e.ID  
117         WHERE t.division_name = e.division_name; -- Match division names  
118  
119 BEGIN  
120     FOR r_trainee IN c_trainees_with_same_division LOOP  
121         DBMS_OUTPUT.PUT_LINE('Trainee Name: ' || r_trainee.trainee_name ||  
122                               ', Division: ' || r_trainee.division_name ||  
123                               ', Mentor Name: ' || r_trainee.mentor_name);  
124     END LOOP;  
125 END;  
126 /  
127  
128
```

Results Explain Describe Saved SQL History

Trainee Name: Jane Doe, Division: Data Science, Mentor Name: Raghav Sharma
Trainee Name: Sophia Martinez, Division: Data Science, Mentor Name: Raghav Sharma
Trainee Name: John Smith, Division: Robotics, Mentor Name: Ankit Verma
Trainee Name: Liam Patel, Division: Robotics, Mentor Name: Ankit Verma
Trainee Name: Olivia Brown, Division: AI Research, Mentor Name: Amit Verma
Trainee Name: Alice Johnson, Division: AI Research, Mentor Name: Amit Verma
Trainee Name: Liam Patel, Division: Robotics, Mentor Name: Sanjay Patel
Trainee Name: Michael Lee, Division: Data Science, Mentor Name: Priya Kumar
Trainee Name: Daniel Clark, Division: AI Research, Mentor Name: Neha Singh
Trainee Name: Liam Patel, Division: Robotics, Mentor Name: Rajesh Gupta
Trainee Name: Olivia Brown, Division: AI Research, Mentor Name: Prof. Suchetana Chakraborty
Trainee Name: Daniel Clark, Division: AI Research, Mentor Name: Prof. Suchetana Chakraborty

Statement processed.

karanganeshtp@gmail.com karan12345 Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.1.3

- 4) Highest Operating Budget: Write a PL/SQL block to find the division_name with the highest operating budget.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a PL/SQL block in the 'Script Editor'. The schema is set to 'WKSP_KARAN12345'. The block is as follows:

```
120  
127  
128 ----- 4 -----  
129  
130 DECLARE  
131     v_highest_budget NUMBER;  
132     v_division_name VARCHAR2(50);  
133 BEGIN  
134     SELECT division_name, operating_budget  
135     INTO v_division_name, v_highest_budget  
136     FROM division  
137     WHERE operating_budget = (  
138         SELECT MAX(operating_budget) FROM division  
139     );  
140  
141     DBMS_OUTPUT.PUT_LINE('Division with the Highest Operating Budget: ' || v_division_name |  
142         ', Budget: ' || v_highest_budget);  
143 EXCEPTION  
144     WHEN NO_DATA_FOUND THEN  
145         DBMS_OUTPUT.PUT_LINE('No divisions found.');
```

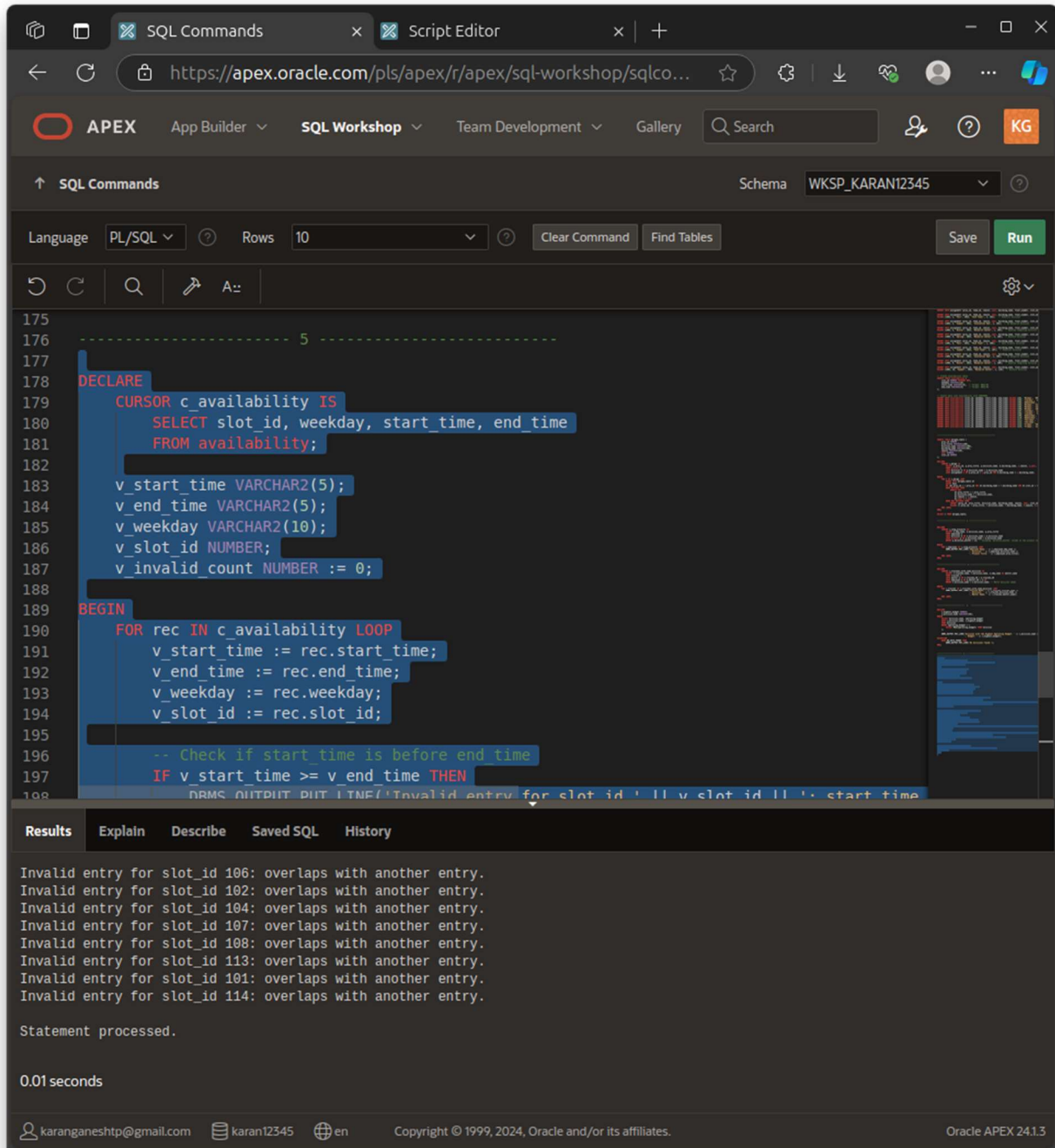
The 'Results' tab is selected, showing the output of the statement:

```
Division with the Highest Operating Budget: Quantum Computing, Budget: 1400000  
Statement processed.  
0.01 seconds
```

The footer of the interface shows the user 'karangeshtp@gmail.com', the schema 'karan12345', the language 'en', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.

- 5) Validate Slot Availability: Write a PL/SQL code to check the validity of the availability entries. A valid slot should have the start_time before the end_time and should not overlap with other entries for the same slot_id and weekday.

Output :



The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' menus. The 'SQL Commands' tab is active, showing a schema of 'WKSP_KARAN12345'. The language is set to 'PL/SQL' and the number of rows is set to '10'. The script editor contains the following PL/SQL code:

```
175 ----- 5 -----
176
177 DECLARE
178     CURSOR c_availability IS
179         SELECT slot_id, weekday, start_time, end_time
180         FROM availability;
181
182     v_start_time VARCHAR2(5);
183     v_end_time VARCHAR2(5);
184     v_weekday VARCHAR2(10);
185     v_slot_id NUMBER;
186     v_invalid count NUMBER := 0;
187
188 BEGIN
189     FOR rec IN c_availability LOOP
190         v_start_time := rec.start_time;
191         v_end_time := rec.end_time;
192         v_weekday := rec.weekday;
193         v_slot_id := rec.slot_id;
194
195         -- Check if start time is before end time
196         IF v_start_time >= v_end_time THEN
197             DBMS_OUTPUT.PUT_LINE('Invalid entry for slot_id ' || v_slot_id || ' : start time
```

The 'Results' tab shows the output of the script:

```
Invalid entry for slot_id 106: overlaps with another entry.
Invalid entry for slot_id 102: overlaps with another entry.
Invalid entry for slot_id 104: overlaps with another entry.
Invalid entry for slot_id 107: overlaps with another entry.
Invalid entry for slot_id 108: overlaps with another entry.
Invalid entry for slot_id 113: overlaps with another entry.
Invalid entry for slot_id 101: overlaps with another entry.
Invalid entry for slot_id 114: overlaps with another entry.

Statement processed.

0.01 seconds
```

The footer of the interface includes the user 'karangeshtp@gmail.com', the schema 'karan12345', the language 'en', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.

- 6) Count Projects by Season: Write a PL/SQL block to count the number of projects each division is working on during a given season (e.g., "Winter").

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a script editor with a PL/SQL block. The script defines variables for season, division name, and project count, then uses a cursor to loop through divisions and count projects for the specified season. The output pane shows the results of the execution.

```
225 ----- 6 -----
226
227 DECLARE
228   v_season VARCHAR2(20) := 'Winter'; -- Specify the season
229   v_division_name VARCHAR2(50);
230   v_project_count NUMBER;
231
232   CURSOR c_projects IS
233     SELECT d.division_name, COUNT(e.proj_id) AS project_count
234     FROM division d
235     JOIN project p ON d.division_name = p.division_name -- Assuming project table has division_name
236     JOIN enrollment e ON p.proj_id = e.proj_id
237     WHERE e.season = v_season
238     GROUP BY d.division_name;
239 BEGIN
240   -- Open the cursor and loop through each division
241   FOR rec IN c_projects LOOP
242     v_division_name := rec.division_name;
243     v_project_count := rec.project_count;
244
245     DBMS_OUTPUT.PUT_LINE('Division: ' || v_division_name || ', Project Count: ' || v_project_count);
246   END LOOP;
247
```

Results | Explain | Describe | Saved SQL | History

Division: Data Science, Project Count: 1
Division: Robotics, Project Count: 1

Statement processed.

0.05 seconds

karangeshtp@gmail.com | karan12345 | en | Copyright © 1999, 2024, Oracle and/or its affiliates. | Oracle APEX 24.1.3

- 7) Update Project IDs: Write a PL/SQL block to update the proj_id in the works_on table to follow the format [First 3 letters of division_name]_[proj_id].

Output :

Since Proj_ID is foreign key in the woks_on table with data type as INT, it can not be changed/formatted in this works_on table because of Intergrity Constraints. Therefore created a new attribute as formatted_proj_id in the works_on Table.

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a PL/SQL block for updating project IDs in the 'works_on' table. The block includes an ALTER TABLE statement to add a 'formatted_proj_id' column, a DECLARE statement for a new project ID variable, a BEGIN statement, a FOR loop to iterate through the 'works_on' table, and an UPDATE statement to set the new project ID. The block concludes with an END LOOP statement and a COMMIT statement. The 'Results' tab is selected, displaying a table with the following data:

ID	PROJ_ID	TEAM_ID	SEASON	YEAR	FORMATTED_PROJ_ID
101	1001	1	Spring	2022	Dat_1001
106	1002	3	Summer	2023	Rob_1002
102	1002	2	Fall	2023	Rob_1002
105	1001	2	Spring	2022	Dat_1001
103	1003	3	Summer	2022	Al_1003
104	1004	1	Winter	2023	Qua_1004

The footer of the interface shows the user 'karanganeshtp@gmail.com', the schema 'karan12345', and the Oracle APEX version '24.1.3'.

- 8) Calculate Total Salary: Write a PL/SQL block to compute the total salary expenditure for a project based on the employees assigned to that project.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a PL/SQL script in the 'Script Editor'. The script is a PL/SQL block that declares a cursor 'c_projects' to select project IDs from the 'project' table. It then loops through each project, calculating the total salary expenditure by summing the annual salaries of employees assigned to that project. The results are outputted using 'DBMS_OUTPUT.PUT_LINE'. The 'Run' button is visible on the right side of the editor.

```
307  
308 ----- 8 -----  
309  
310 DECLARE  
311     CURSOR c_projects IS  
312         SELECT proj_id  
313             FROM project; -- Get all project IDs  
314  
315     v_total_salary NUMBER; -- Variable to store the total salary for each project  
316 BEGIN  
317     -- Loop through each project  
318     FOR rec IN c_projects LOOP  
319         -- Calculate total salary for the current project  
320         SELECT NVL(SUM(e.annual_salary), 0) INTO v_total_salary  
321             FROM employee e  
322             JOIN works_on w ON e.ID = w.ID  
323             WHERE w.proj_id = rec.proj_id;  
324  
325         -- Output the total salary for the current project  
326         DBMS_OUTPUT.PUT_LINE('Total salary expenditure for project ID ' || rec.proj_id || '  
327     END LOOP;  
328 EXCEPTION  
329     WHEN OTHERS THEN  
330         DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
```

The 'Results' tab shows the output of the PL/SQL block execution:

```
Total salary expenditure for project ID 1001 is: 177000  
Total salary expenditure for project ID 1002 is: 183000  
Total salary expenditure for project ID 1003 is: 190000  
Total salary expenditure for project ID 1004 is: 175000  
Total salary expenditure for project ID 1005 is: 250000  
  
Statement processed.  
  
0.03 seconds
```

The footer of the interface includes the user 'karanganeshtp@gmail.com', the schema 'karan12345', the language 'en', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.

- 9) List Employees on Summer Projects: Write a PL/SQL block to output the first name, last name, and division_name of all employees who have been assigned to a project during the "Summer" season for the current year.

Output :

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a PL/SQL block for the schema 'WKSP_KARAN12345'. The block is as follows:

```
341 ----- 9 -----
342 DECLARE
343     CURSOR emp_cursor IS
344         SELECT e.emp_name, e.division_name
345         FROM employee e
346         JOIN works_on w ON e.ID = w.ID
347         WHERE w.season = 'Summer' AND w.year = 2024;
348
349     emp_rec emp_cursor%ROWTYPE;
350     first_name VARCHAR2(100);
351     last_name VARCHAR2(100);
352 BEGIN
353     FOR emp_rec IN emp_cursor LOOP
354         -- Find the position of the space between first and last names
355         DECLARE
356             space_pos NUMBER;
357         BEGIN
358             space_pos := INSTR(emp_rec.emp_name, ' '); -- Find the first space
359
360             -- Ensure that the space is found to avoid errors
361             IF space_pos > 0 THEN
362                 -- Extract first and last names
363                 first_name := SUBSTR(emp_rec.emp_name, 1, space_pos - 1);
```

The 'Results' tab shows the output of the query:

First Name:	Raghav	Last Name:	Sharma	Division:	Data Science
First Name:	Ankit	Last Name:	Verma	Division:	Robotics
First Name:	Amit	Last Name:	Verma	Division:	AI Research
First Name:	Sanjay	Last Name:	Patel	Division:	Robotics
First Name:	Priya	Last Name:	Kumar	Division:	Data Science

Statement processed.
0.02 seconds

The footer shows the user 'karangeshtp@gmail.com', schema 'karan12345', language 'en', copyright '© 1999, 2024, Oracle and/or its affiliates.', and version 'Oracle APEX 24.1.3'.

10) Most Frequent Division in Projects: Write a PL/SQL block to find the division that has the most number of projects assigned to it and list the division_name and proj_title.

Output:

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes the APEX logo, App Builder, SQL Workshop, Team Development, and Gallery. The current schema is WKSP_KARAN12345. The SQL Commands section shows a PL/SQL script with the following code:

```
379 ----- 10 -----
380
381 DECLARE
382     CURSOR division_cursor IS
383         SELECT d.division_name, COUNT(p.proj_id) AS project_count
384         FROM division d
385         LEFT JOIN project p ON d.division_name = p.division_name
386         GROUP BY d.division_name
387         ORDER BY project_count DESC;
388
389     most_frequent_division division_cursor%ROWTYPE;
390     max_projects NUMBER := 0;
391 BEGIN
392     -- Fetch the division with the most projects
393     OPEN division_cursor;
394     FETCH division_cursor INTO most_frequent_division;
395
396     -- Check if any divisions are found
397     IF division_cursor%FOUND THEN
398         max_projects := most_frequent_division.project_count;
399         DBMS_OUTPUT.PUT_LINE('Most Frequent Division: ' || most_frequent_division.division_n
400     END IF;
401
```

The Results section shows the output of the script:

```
Most Frequent Division: Quantum Computing with 3 projects.
Project Title: Cloud Migration
Project Title: Quantum Circuit Design
Project Title: Cyber Security

Statement processed.

0.02 seconds
```

The bottom status bar shows the user karangeshtp@gmail.com, the schema karan12345, and the Oracle APEX 24.1.3 version.

11) Trigger on Employee Update: Write a PL/SQL trigger that displays all details of an employee whenever their salary is updated in the employee table.

Output:

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery' menus, along with a search bar and a user profile icon labeled 'KG'. The main header shows 'SQL Commands' and the selected schema 'WKSP_KARAN12345'. Below this, a toolbar contains options for 'Language' (set to PL/SQL), 'Rows' (set to 10), 'Clear Command', 'Find Tables', 'Save', and 'Run'. The central editor area shows a PL/SQL script with line numbers 408 to 431. The script includes a loop for project details and a trigger definition 'trg_employee_salary_update' that outputs employee details after a salary update. The bottom section, titled 'Results', shows the message 'Trigger created.' and the execution time '0.05 seconds'. The footer contains the user email 'karangeshtp@gmail.com', the schema name 'karan12345', a globe icon with 'en', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.

```
408 WHERE division_name = most_frequent_division.division_name; LOOP
409     DBMS_OUTPUT.PUT_LINE('Project Title: ' || proj_rec.proj_title);
410 END LOOP;
411 END;
412 /
413
414 ----- 11 -----
415
416 CREATE OR REPLACE TRIGGER trg_employee_salary_update
417 AFTER UPDATE OF annual_salary ON employee
418 FOR EACH ROW
419 BEGIN
420     -- Display employee details after salary update
421     DBMS_OUTPUT.PUT_LINE('Employee Details Updated:');
422     DBMS_OUTPUT.PUT_LINE('ID: ' || :NEW.ID);
423     DBMS_OUTPUT.PUT_LINE('Name: ' || :NEW.emp_name);
424     DBMS_OUTPUT.PUT_LINE('Division: ' || :NEW.division_name);
425     DBMS_OUTPUT.PUT_LINE('New Salary: ' || :NEW.annual_salary);
426     DBMS_OUTPUT.PUT_LINE('-----');
427 END;
428 /
429
430
431
```

Results Explain Describe Saved SQL History

Trigger created.

0.05 seconds

karangeshtp@gmail.com karan12345 en Copyright © 1999, 2024, Oracle and/or its affiliates. Oracle APEX 24.1.3

12) Insert Projects Using Cursor: Insert two new projects in the project table (e.g., “Cyber Security”, “Cloud Migration”) with arbitrary attributes. Now, using a CURSOR, write a PL/SQL code to enter a new record in the dependency table corresponding to the projects listed in project that do not have any dependencies in the dependency table.

Output:

The screenshot displays the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' tab is active, showing a script editor with the following PL/SQL code:

```
430
431
432 ----- 12 -----
433
434 -- Insert new projects into the project table
435 INSERT INTO project (proj_id, proj_title, division_name, duration_months) VALUES (1006, 'Cyber Security', 'IT', 12);
436 INSERT INTO project (proj_id, proj_title, division_name, duration_months) VALUES (1007, 'Cloud Migration', 'IT', 12);
437
438 DECLARE
439     CURSOR proj_cursor IS
440         SELECT proj_id, proj_title
441         FROM project
442         WHERE proj_id NOT IN (SELECT proj_id FROM dependency);
443
444     v_proj_id project.proj_id%TYPE;
445     v_proj_title project.proj_title%TYPE;
446
447 BEGIN
448     -- Loop through projects without dependencies
449     FOR proj_record IN proj_cursor LOOP
450         v_proj_id := proj_record.proj_id;
451         v_proj_title := proj_record.proj_title;
452         INSERT INTO dependency (proj_id, dependent_proj_id)
453             VALUES (v_proj_id, 1001);
454     END LOOP;
455 END;
```

The 'Results' tab at the bottom shows the execution output:

```
Inserted dependency for project: Cloud Migration with dependency on project ID: 1001
Inserted dependency for project: Cyber Security with dependency on project ID: 1001

1 row(s) inserted.

0.04 seconds
```

The footer of the interface includes the user email 'karanganeshtp@gmail.com', the schema 'karan12345', the language 'en', the copyright notice 'Copyright © 1999, 2024, Oracle and/or its affiliates.', and the version 'Oracle APEX 24.1.3'.