

+

**Project Report
Submitted for**

DATABASE MANAGEMENT SYSTEM-(UCS310)

PROJECT MANAGEMENT SYSTEM

Submitted by:

(102003447) TANISHA PARKASH

(102003448) ARUSHI

(102003450) SMITI SINGH

(102003453) VIKUL VERMA

(102003460) SUNIDHI SURI

BE Second Year

Submitted to

Mrs. Shubhani Aggarwal



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

TIET, Patiala

Jan-June 2022

TABLE OF CONTENTS

ABSTRACT	i
DECLARATION	ii
TABLE OF CONTENTS	iii

ABSTRACT

It is very tedious to find and manage projects and research papers using manual or classical processes. The main goal of this project is to build an integrated framework to help the users find projects that suit their needs. Project management system is a system for the management and supervision of projects and research papers. It is a management system that is useful to students, project managers, as well as professors.

DECLARATION

We, the undersigned, solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of Ms. Shubhangi Aggarwal

We assert that the statements made and conclusions drawn are an outcome of our research work. We further certify that-

I. The work contained in the report is original and has been done by us under the general supervision of our supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or the any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and given their details in the references.

<i>TANISHA PARKASH</i>	<i>102003447</i>
<i>ARUSHI</i>	<i>102003448</i>
<i>SMITI SINGH</i>	<i>102003450</i>
<i>VIKUL VERMA</i>	<i>102003453</i>
<i>SUNIDHI SURI</i>	<i>102003460</i>

INDEX

S. No.	
1.	Introduction
3.	Problem Statement
4.	ER Diagram
5.	ER Diagram to ER Tables
6.	Normalized Tables
7.	Implementation
8.	PL/SQL Implementation
9.	Working Screenshots

PROBLEM STATEMENT

In the proposed project we will develop a system to manage the ongoing/completed projects in the university. Students and professors can make their profiles on the system, and provide details of their specialization. Accordingly, the system can help them find a project that suits their needs. The users are able to obtain information about relevant projects/research papers and form teams.

OVERVIEW

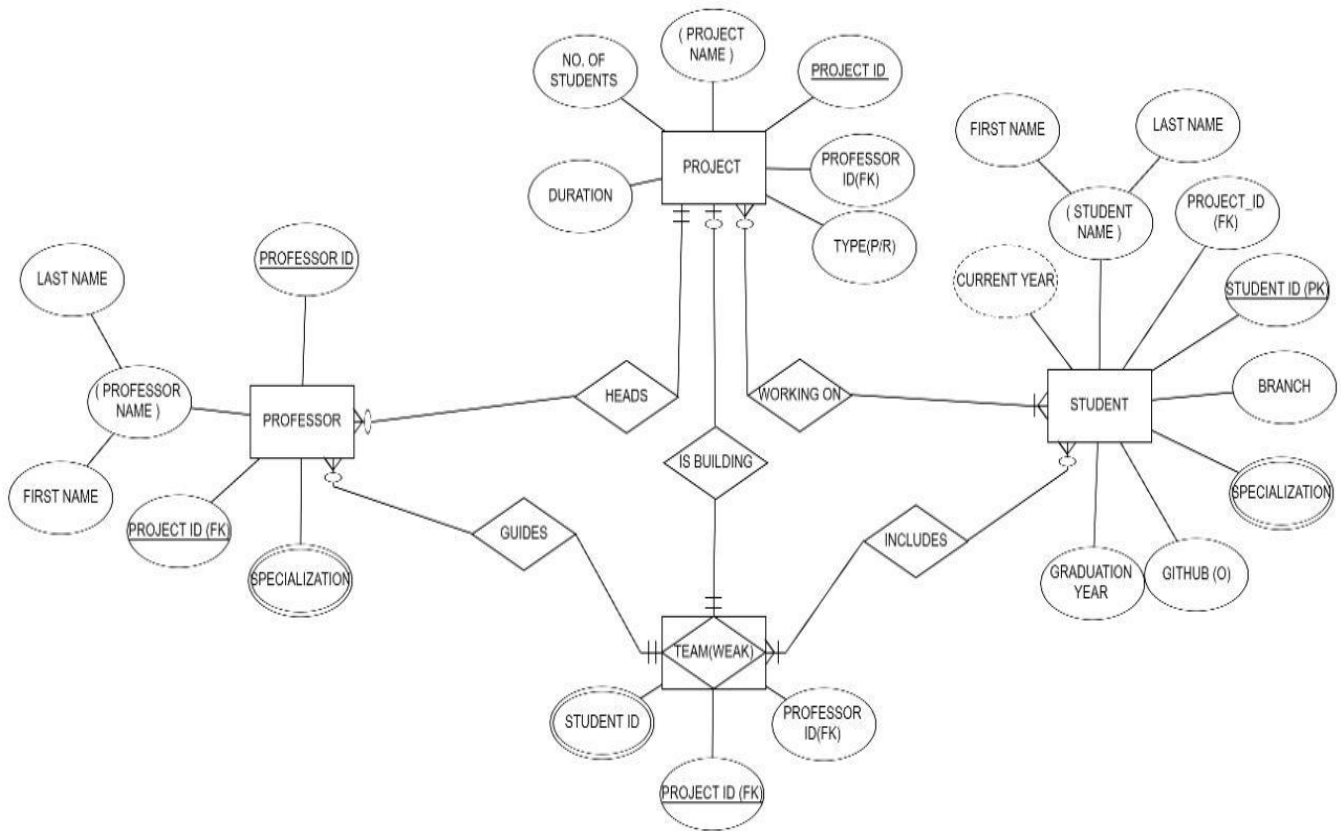
Project-based learning not only provides opportunities for students to collaborate or drive their own learning, but it also teaches them skills such as problem solving, and helps to develop additional skills integral to their future, such as critical thinking and time management.

However, it is burdensome to look for projects/ research papers that match ones interests using conventional methods. This integrated system makes finding projects/research papers significantly more efficient.

Every user can register to database by providing a brief information about themselves. This project also allows the users to filter ongoing projects that match their specialization. Teams can also look for members that have the required skills needed for their projects. Professors can find collaborators for their research papers as well.

We have used Oracle Live SQL for the implementation of this project.

ER DIAGRAM



ER TO TABLES

```
create table Professor(Prof_ID number(5) primary key, Prof_name char(200) NOT NULL, Dept char(100),
```

```
Project_ID number(5) UNIQUE, Specialization varchar(500));
```

```
alter table Professor add constraint c1 check (Dept in('CSED', 'ECED', 'MECH', 'BIOTECH', 'IT', 'INDUSTRIAL', 'CHEMICAL'));
```

```
desc professor;
```

SQL Worksheet

Clear Find Actions Save Run

```
1 create table Professor(Prof_ID number(5) primary key, Prof_name char(200) NOT NULL, Dept char(100),
2 Project_ID number(5) UNIQUE, Specialization varchar(500));
3 alter table Professor add constraint c1 check (Dept in('CSED', 'ECED', 'MECH', 'BIOTECH', 'IT', 'INDUSTRIAL', 'CHEMICAL'));
4 desc professor;
5
6
7
```

Table created.

Table altered.

TABLE PROFESSOR

Column	Null?	Type
PROF_ID	NOT NULL	NUMBER(5,0)
PROF_NAME	NOT NULL	CHAR(200)
DEPT	-	CHAR(100)
PROJECT_ID	-	NUMBER(5,0)
SPECIALIZATION	-	VARCHAR2(500)

Download CSV
5 rows selected.

```
create table Project(Project_ID number(5) primary key, Proj_name char(200) NOT NULL, Prof_ID number(5) UNIQUE, Type char(2), Team_name char(100) NOT NULL,
```

```
Start_date date, Duration_in_months number(3), Status char(100));
```

```
alter table Project add constraint c2 check(Type in('P','R'));
```

```
alter table Project add constraint c3 check(Status in('IN PROGRESS', 'COMPLETED'));
```

```
desc project;
```


SQL Worksheet

Clear Find Actions Save Run

```

5
6 create table Project(Project_ID number(5) primary key, Proj_name char(200) NOT NULL, Prof_ID number(5) UNIQUE, Type char(2), Team_name char(100) NOT NULL,
7 Start_date date, Duration_in_months number(3), Status char(100));
8 alter table Project add constraint c2 check(Type in('P', 'R'));
9 alter table Project add constraint c3 check(Status in('IN PROGRESS', 'COMPLETED'));
10 desc project;
11
12 create table Student (Roll_no number(10) primary key, S_name char(200) NOT NULL, Branch char(5), Specialization varchar(500), Grad_year number(4), Curr_year number(2), Project_ID number(5));
13
14

```

Table altered.

TABLE PROJECT

Column	Null?	Type
PROJECT_ID	NOT NULL	NUMBER(5,0)
PROJ_NAME	NOT NULL	CHAR(200)
PROF_ID	-	NUMBER(5,0)
TYPE	-	CHAR(2)
TEAM_NAME	NOT NULL	CHAR(100)
START_DATE	-	DATE
DURATION_IN_MONTHS	-	NUMBER(3,0)
STATUS	-	CHAR(100)

Download CSV

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym

create table Student (Roll_no number(10) primary key, S_name char(200) NOT NULL, Branch char(5), Specialization varchar(500), Grad_year number(4), Curr_year number(2), Project_ID number(5));

desc student;

SQL Worksheet

Clear Find Actions Save Run

```

10 desc project;
11
12 create table Student (Roll_no number(10) primary key, S_name char(200) NOT NULL, Branch char(5), Specialization varchar(500), Grad_year number(4), Curr_year number(2), Project_ID number(5));
13 desc student;
14
15 create table Team(Team_Name char(200) primary key, Proj_name char(200), Prof_name char(200), No_of_students number(4));
16
17
18
19

```

Table created.

TABLE STUDENT

Column	Null?	Type
ROLL_NO	NOT NULL	NUMBER(10,0)
S_NAME	NOT NULL	CHAR(200)
BRANCH	-	CHAR(5)
SPECIALIZATION	-	VARCHAR2(500)
GRAD_YEAR	-	NUMBER(4,0)
CURR_YEAR	-	NUMBER(2,0)
PROJECT_ID	-	NUMBER(5,0)

Download CSV

7 rows selected.

create table Team(Team_Name char(200) primary key, Proj_name char(200), Prof_name char(200), No_of_students number(4));

desc team;

SQL Worksheet

ClearFindActionsSaveRun

```
10 desc project;
11
12 create table Student (Roll_no number(10) primary key, S_name char(200) NOT NULL, Branch char(5), Specialization varchar(500), Grad_year number(4), Curr_year number(2), Project_ID number(5));
13 desc student;
14
15 create table Team(Team_Name char(200) primary key, Proj_name char(200), Prof_name char(200), No_of_students number(4));
16 desc team;
17
18
19
```

Table created.

TABLE TEAM

Column	Null?	Type
TEAM_NAME	NOT NULL	CHAR(200)
PROJ_NAME	-	CHAR(200)
PROF_NAME	-	CHAR(200)
NO_OF_STUDENTS	-	NUMBER(4,0)

Download CSV

4 rows selected.

NORMALISATION

- **1-NF** - We converted every attribute in a relation to a singled valued attribute. Our model contained 2 multivalued attributes in the Student and Professor table respectively - We removed this redundancy by creating two more tables i.e. 'STUDENT - SPECIALIZATION' table and 'PROFESSOR-SPECIALIZATION' table. Further, we converted the 'NAME' attribute to 'FIRST-NAME' and 'SECOND-NAME' since it was a composite attribute.
- **2-NF** - We observed that all the tables were already in 2-NF as each table only one simple primary key.
- **3-NF** - We observed that student and project tables had **transitive dependency** as project id could determine team name. So, we removed the unnecessary columns from the table, that caused transitive dependency.
- **4-NF** - We observed that all the tables were already in 4-NF.
- **5-NF** - We observed that all the tables were already in 5-NF.

SQL Worksheet

Clear

Find

Actions

Save

Run

```
27
28 --normalisation
29 --student table ka child: student_specialization table
30 create table S_specialization(Roll_no number(10) , Specialization char(50));
31 alter table Student drop column Specialization;
32 alter table S_specialization add constraint f5 foreign key (Roll_no) references Student (Roll_no);
33 alter table Student drop column S_name;
34 alter table Student add S_firstname char(50) NOT NULL;
35 alter table Student add S_lastname char(50);
36 --student table is in Inf
37
38 create table P_specialization(Prof_ID number(5), Specialization char(50));
39 alter table Professor drop column Specialization;
40 alter table Professor add P_firstname char(50) NOT NULL;
41 alter table Professor add P_lastname char(50);
42
```

Table created.

Table altered.

Table altered.

Table altered.

Table altered.

Table altered.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym

SQL Worksheet

Clear

Find

Actions

Save

Run

```
35 alter table Student add S_lastname char(50);
36 --student table is in Inf
37
38 create table P_specialization(Prof_ID number(5), Specialization char(50));
39 alter table Professor drop column Specialization;
40 alter table Professor add P_firstname char(50) NOT NULL;
41 alter table Professor add P_lastname char(50);
42 alter table Professor drop column Prof_name;
43 --professor table in Inf
44
45 --project table already in Inf
46 alter table Team drop column Prof_name;
47 --team table in Inf
48
49
50
```

Table created.

Table altered.

Table altered.

Table altered.

Table altered.

Table altered.

Table altered.

SQL Worksheet

Clear

Find

Actions

Save

Run

```
47 --team table in Inf
48
49 --all tables already in 2nf as each has only one primary key
50
51 --student table has transitive dependency as a project id can determine team name so it can appear as a separate table
52 desc Project
53 desc Team
54 desc Professor
55 desc Student
56
57 alter table Student drop constraint f4;
58 alter table Student drop column Team_name;
59
60
61 --DATA INSERTION
62
```

Table altered.

Table altered.

IMPLEMENTATION

--normalisation

--student table ka child: student_specialization table

```
create table S_specialization(Roll_no number(10) ,  
Specialization char(50));
```

```
alter table Student drop column Specialization;
```

```
alter table S_specialization add constraint f5 foreign key  
(Roll_no) references Student (Roll_no);
```

```
alter table Student drop column S_name;
```

```
alter table Student add S_firstname char(50) NOT NULL;
```

```
alter table Student add S_lastname char(50);
```

--student table is in 1nf

```
create table P_specialization(Prof_ID number(5),  
Specialization char(50));
```

alter table Professor drop column Specialization;

alter table Professor add P_firstname char(50) NOT NULL;

alter table Professor add P_lastname char(50);

alter table Professor drop column Prof_name;

--professor table in 1nf

--project table already in 1nf

alter table Team drop column Prof_name;

--team table in 1nf

--all tables already in 2nf as each has only one primary key

--student table has transitive dependency as a project id can determine team name so it can appear as a separate table

desc Project

desc Team

```
desc Professor
```

```
desc STudent
```

```
alter table Student drop constraint f4;
```

```
alter table Student drop column Team_name;
```

DATA INSERTION

```
insert into Team values('SEIMMENS', 'VOICE_ASSISTANT', 3);
```

```
INSERT INTO TEAM VALUES('FRONTIERS',  
'LOGIN_AUTHENTICATION', 5);
```

```
INSERT INTO TEAM VALUES('XBEEES', 'LINE_FOLLOWER', 5);
```

```
INSERT INTO TEAM VALUES('CAFFEINE', 'SaaS_WEBPAGE',4);
```

```
INSERT INTO TEAM VALUES('PHOENIX',  
'LIBRARY_MANAGEMENT_SYSTEM', 4);
```

```
INSERT INTO TEAM VALUES('STARS', 'RESTAURANT_APP',5);
```

```
INSERT INTO TEAM VALUES('INNOVATORS',  
'ENRON_INVESTIGATION',4);
```

```
INSERT INTO TEAM VALUES('ALLIES',  
'STOCK_PREDICTION',3);
```

```
INSERT INTO TEAM VALUES('LEGENDS',  
'SPACE_SHOOTER_GAME',4);
```

```
INSERT INTO TEAM  
VALUES('CHAMPIONS','AUTOMATED_CAR',4);
```


select *from team;

SQL Worksheet

Clear Find Actions Save

```
--DATA INSERTION
61 insert into team values('SEIWIENS', 'VOICE_ASSISTANT', 3);
62 INSERT INTO TEAM VALUES('FRONTIERS', 'LOGIN_AUTHENTICATION', 5);
63 INSERT INTO TEAM VALUES('XBEEES', 'LINE_FOLLOWER', 5);
64 INSERT INTO TEAM VALUES('CAFFEINE', 'SaaS_WEBPAGE', 4);
65 INSERT INTO TEAM VALUES('PHOENIX', 'LIBRARY_MANAGEMENT_SYSTEM', 4);
66 INSERT INTO TEAM VALUES('STARS', 'RESTAURANT_APP', 5);
67 INSERT INTO TEAM VALUES('INNOVATORS', 'ENRON_INVESTIGATION', 4);
68 INSERT INTO TEAM VALUES('ALLIES', 'STOCK_PREDICTION', 3);
69 INSERT INTO TEAM VALUES('LEGENDS', 'SPACE_SHOOTER_GAME', 4);
70 INSERT INTO TEAM VALUES('CHAMPIONS', 'AUTOMATED_CAR', 4);
71
72 select *from team;
```

TEAM_NAME	PROJ_NAME	NO_OF_STU
SEIWIENS	VOICE_ASSISTANT	3
FRONTIERS	LOGIN_AUTHENTICATION	5
XBEEES	LINE_FOLLOWER	5
CAFFEINE	SaaS_WEBPAGE	4
PHOENIX	LIBRARY_MANAGEMENT_SYSTEM	4
STARS	RESTAURANT_APP	5
INNOVATORS	ENRON_INVESTIGATION	4
ALLIES	STOCK_PREDICTION	3

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX - Privacy - Terms of Use

insert into professor values(1001, 'CSED', 'GEETA', 'KASANA');

insert into professor values(1002, 'ECED', 'RAVINDRA',
'KUMAR');

insert into professor values(1003, 'MECH', 'KAMAL', 'SINGH');

insert into professor values(1004, 'BIOTECH', 'SHALINI',
'SHARMA');

insert into professor values(1005, 'IT', 'SHARAD', 'SAXENA');

SELECT* FROM PROFESSOR;

SQL Worksheet

Clear Find Actions Save

```
75 REVOKE TABLE PROFESSOR DROP CONSTRAINT P1;
76 ALTER TABLE PROFESSOR DROP COLUMN Project_ID;
77
78 insert into professor values(1001, 'CSED', 'GEETA', 'KASANA');
79 insert into professor values(1002, 'ECED', 'RAVINDRA', 'KUMAR');
80 insert into professor values(1003, 'MECH', 'KAMAL', 'SINGH');
81 insert into professor values(1004, 'BIOTECH', 'SHALINI', 'SHARMA');
82 insert into professor values(1005, 'IT', 'SHARAD', 'SAXENA');
83
84 SELECT* FROM PROFESSOR;
85
86 INSERT INTO PROJECT VALUES (3001, 'STOCK_PREDICTION', 1001, 'P', 'ALLIES', TO_DATE('17-02-2022','DD-MM-YYYY'), 4, 'IN PROGRESS');
87 INSERT INTO PROJECT VALUES (3002, 'LIBRARY_MANAGEMENT_SYSTEM', 1002, 'P', 'PHOENIX', TO_DATE('16-05-2022','DD-MM-YYYY'), 6, 'IN PROGRESS');
88 INSERT INTO PROJECT VALUES (3003, 'RESTAURANT_APP', 1003, 'P', 'STARS', TO_DATE('20-08-2020','DD-MM-YYYY'), 5, 'IN PROGRESS');
89 INSERT INTO PROJECT VALUES (3004, 'AUTOMATED_CAR', 1004, 'P', 'CHAMPIONS', TO_DATE('09-04-2021','DD-MM-YYYY'), 2, 'IN PROGRESS');
90
```

Download CSV
5 rows selected.

PROF_ID	DEPT	P_FIRSTNAME	P_LASTNAME
1001	CSED	GEETA	KASANA
1002	ECED	RAVINDRA	KUMAR
1003	MECH	KAMAL	SINGH
1004	BIOTECH	SHALINI	SHARMA
1005	IT	SHARAD	SAXENA

INSERT INTO PROJECT VALUES (3001, 'STOCK_PREDICTION',
1001, 'P', 'ALLIES', TO_DATE('17-02-2022','DD-MM-YYYY'), 4,
'IN PROGRESS');

INSERT INTO PROJECT VALUES (3002,
'LIBRARY_MANAGEMENT_SYSTEM', 1002, 'P', 'PHOENIX',
TO_DATE('16-05-2022','DD-MM-YYYY'), 6, 'IN PROGRESS');

INSERT INTO PROJECT VALUES (3003, 'RESTAURANT_APP',
1003, 'P', 'STARS', TO_DATE('20-08-2020','DD-MM-YYYY'), 5, 'IN
PROGRESS');

```
INSERT INTO PROJECT VALUES (3004, 'AUTOMATED_CAR',  
1004, 'P', 'CHAMPIONS',
```

```
TO_DATE('09-04-2021','DD-MM-YYYY'), 2, 'IN PROGRESS');
```

```
INSERT INTO PROJECT VALUES (3005, 'SaaS_WEBPAGE',  
1005, 'P', 'CAFFEINE', TO_DATE('25-06-2020','DD-MM-YYYY'),  
6, 'IN PROGRESS');
```

```
DESC PROJECT
```

```
SELECT * FROM PROJECT
```

SQL Worksheet

Clear Find Actions Save

```
83  
84 SELECT * FROM PROFESSOR;  
85  
86 INSERT INTO PROJECT VALUES (3001, 'STOCK_PREDICTION', 1001, 'P', 'ALLIES', TO_DATE('17-02-2022','DD-MM-YYYY'), 4, 'IN PROGRESS');  
87 INSERT INTO PROJECT VALUES (3002, 'LIBRARY_MANAGEMENT_SYSTEM', 1002, 'P', 'PHOENIX', TO_DATE('16-05-2022','DD-MM-YYYY'), 6, 'IN PROGRESS');  
88 INSERT INTO PROJECT VALUES (3003, 'RESTAURANT_APP', 1003, 'P', 'STARS', TO_DATE('20-08-2020','DD-MM-YYYY'), 5, 'IN PROGRESS');  
89 INSERT INTO PROJECT VALUES (3004, 'AUTOMATED_CAR', 1004, 'P', 'CHAMPIONS', TO_DATE('09-04-2021','DD-MM-YYYY'), 2, 'IN PROGRESS');  
90 INSERT INTO PROJECT VALUES (3005, 'SaaS_WEBPAGE', 1005, 'P', 'CAFFEINE', TO_DATE('25-06-2020','DD-MM-YYYY'), 6, 'IN PROGRESS');  
91 DESC PROJECT  
92 SELECT * FROM PROJECT  
93  
94 INSERT INTO STUDENT VALUES(102007584,'COE',2024,2,3001,'RAHUL','KADIYAN');  
95 INSERT INTO STUDENT VALUES(102007560,'ENC',2025,1,3002,'PREH','KAPOOR');  
96 INSERT INTO STUDENT VALUES(102007550,'EIC',2023,3,3003,'ROHAN','SINGH');  
97 TRUNCATE TABLE STUDENT;  
98
```

0 rows selected.

PROJECT_ID	PROJ_NAME	PROF_ID	TYPE	TEAM_NAME	START_DATE	DURATION_IN_MONTHS	STATUS
3001	STOCK_PREDICTION	1001	P	ALLIES	17-FEB-22	4	IN PROGRESS
3002	LIBRARY_MANAGEMENT_SYSTEM	1002	P	PHOENIX	16-MAY-22	6	IN PROGRESS
3003	RESTAURANT_APP	1003	P	STARS	20-AUG-20	5	IN PROGRESS
3004	AUTOMATED_CAR	1004	P	CHAMPIONS	09-APR-21	2	IN PROGRESS
3005	SaaS_WEBPAGE	1005	P	CAFFEINE	25-JUN-20	6	IN PROGRESS

Download CSV
5 rows selected.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.0.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle ADEV - Driven - Tame of Fire

```
INSERT INTO STUDENT  
VALUES(102007584,'COE',2024,2,3001,'RAHUL','KADIYAN');
```

```

INSERT INTO STUDENT
VALUES(102007560,'ENC',2025,1,3002,'PREM','KAPOOR');

INSERT INTO STUDENT
VALUES(102007550,'EIC',2023,3,3003,'ROHAN','SINGH');

INSERT INTO STUDENT
VALUES(102007556,'ECE',2024,2,3004,'SIMRAN','MALHOTRA')
;

INSERT INTO STUDENT
VALUES(102007578,'COE',2023,3,3005,'TANISHA','SHARMA');

SELECT * FROM STUDENT

```

SQL Worksheet

Clear Find Actions Save Run

```

90 INSERT INTO PROJECT VALUES (3005, 'SaaS_WEBPAGE', 1005, 'P', 'CAFFEINE', TO_DATE('25-06-2020','DD-MM-YYYY'), 6, 'IN PROGRESS');
91 DESC PROJECT
92 SELECT * FROM PROJECT
93
94 INSERT INTO STUDENT VALUES(102007584,'COE',2024,2,3001,'RAHUL','KADIYAN');
95 INSERT INTO STUDENT VALUES(102007560,'ENC',2025,1,3002,'PREM','KAPOOR');
96 INSERT INTO STUDENT VALUES(102007550,'EIC',2023,3,3003,'ROHAN','SINGH');
97 INSERT INTO STUDENT VALUES(102007556,'ECE',2024,2,3004,'SIMRAN','MALHOTRA');
98 INSERT INTO STUDENT VALUES(102007578,'COE',2023,3,3005,'TANISHA','SHARMA');
99 SELECT * FROM STUDENT
100
101 INSERT INTO P_specialization VALUES(1001,'Dev-ops');
102 INSERT INTO P_specialization VALUES(1001,'Computer_Vision');
103 INSERT INTO P_specialization VALUES(1003,'VLSI_Design');
104
105
106

```

1 row(s) selected.

ROLL_NO	BRANCH	GRAD_YEAR	CURR_YEAR	PROJECT_ID	S_FIRSTNAME	S_LASTNAME
102007584	COE	2024	2	3001	RAHUL	KADIYAN
102007560	ENC	2025	1	3002	PREM	KAPOOR
102007550	EIC	2023	3	3003	ROHAN	SINGH
102007556	ECE	2024	2	3004	SIMRAN	MALHOTRA
102007578	COE	2023	3	3005	TANISHA	SHARMA

Download CSV
5 rows selected.

```

INSERT INTO P_specialization VALUES(1001,'Dev-ops');

```

INSERT INTO P_specialization VALUES(1001,'Computer_Vision');

INSERT INTO P_specialization VALUES(1003,'VLSI_Design');

INSERT INTO P_specialization VALUES(1004,'Machining');

INSERT INTO P_specialization VALUES(1005,'Gene-Mutation');

INSERT INTO P_specialization VALUES(1002,'Machine_learning');

INSERT INTO P_specialization VALUES(1001,'DNA_Analysis');

SELECT * FROM P_specialization

SQL Worksheet

Clear Find Actions Save Run

```
97 INSERT INTO STUDENT VALUES(102007556,'ECE',2024,2,3004,'SIRRAH','NALHOTRA');
98 INSERT INTO STUDENT VALUES(102007576,'CDE',2023,3,3005,'TANISHA','SHARMA');
99 SELECT * FROM STUDENT
100
101 INSERT INTO P_specialization VALUES(1001,'Dev-ops');
102 INSERT INTO P_specialization VALUES(1001,'Computer_Vision');
103 INSERT INTO P_specialization VALUES(1003,'VLSI_Design');
104 INSERT INTO P_specialization VALUES(1004,'Machining');
105 INSERT INTO P_specialization VALUES(1005,'Gene-Mutation');
106 INSERT INTO P_specialization VALUES(1002,'Machine_learning');
107 INSERT INTO P_specialization VALUES(1001,'DNA_Analysis');
108 SELECT * FROM P_specialization
109
110 INSERT INTO S_specialization VALUES(102007584,'FULL_STACK');
111
```

PROF_ID	SPECIALIZATION
1001	Dev-ops
1001	Computer_Vision
1003	VLSI_Design
1004	Machining
1005	Gene-Mutation
1002	Machine_learning
1001	DNA_analysis

Download CSV

INSERT INTO S_specialization VALUES(102007584,'FULL_STACK');

INSERT INTO S_specialization VALUES(102007584,'AI');

```
INSERT INTO S_specialization
VALUES(102007560,'UI/UX_DESIGN');
```

```
INSERT INTO S_specialization
VALUES(102007560,'VLSI_DESIGN');
```

```
INSERT INTO S_specialization
VALUES(102007550,'Gene-Mutation');
```

```
INSERT INTO S_specialization VALUES(102007556,'DEVOPS');
```

```
INSERT INTO S_specialization VALUES(102007556,'Machining');
```

```
INSERT INTO S_specialization
VALUES(102007556,'COMPUTER_VISION');
```

```
SELECT * FROM S_specialization
```

QL Worksheet

Clear Find Actions Save Run

```
109
110 INSERT INTO S_specialization VALUES(102007584,'FULL_STACK');
111 INSERT INTO S_specialization VALUES(102007584,'AI');
112 INSERT INTO S_specialization VALUES(102007560,'UI/UX_DESIGN');
113 INSERT INTO S_specialization VALUES(102007560,'VLSI_DESIGN');
114 INSERT INTO S_specialization VALUES(102007550,'Gene-Mutation');
115 INSERT INTO S_specialization VALUES(102007556,'DEVOPS');
116 INSERT INTO S_specialization VALUES(102007556,'Machining');
117 INSERT INTO S_specialization VALUES(102007556,'COMPUTER_VISION');
118 SELECT * FROM S_specialization
119
120 --PLSQL
121 --PROCEDURES
122 CREATE OR REPLACE PROCEDURE INSERT_STUDENT(v_rolino IN student.Roll_no%type, v_fname IN student.s_firstname%type, v_lname IN student.s_lastname%type, v_branch IN student.branch%type, v_gradyr IN student.gradyr%type)
123
124
```

ROLL_NO	SPECIALIZATION
102007584	FULL_STACK
102007584	AI
102007560	UI/UX_DESIGN
102007560	VLSI_DESIGN
102007550	Gene-Mutation
102007556	DEVOPS
102007556	Machining
102007556	COMPUTER_VISION

Download CSV

PL/SQL IMPLEMENTATION

PROCEDURES

CREATE OR REPLACE PROCEDURE

INSERT_STUDENT(v_rollno IN student.Roll_no%type, v_fname
IN student.s_firstname%type, v_lname IN
student.s_lastname%type, v_branch IN student.branch%type,
v_gradyr IN student.grad_year%type, v_curryr IN
student.curr_year%type, v_projid IN student.project_id%type)

is

begin

insert into student values(v_rollno,v_branch,v_gradyr, v_curryr,
v_projid, v_lname, v_fname);

end insert_student;

create or replace procedure insert_professor(v_profid IN
professor.prof_id%type, v_dept IN professor.dept%type,

```
v_fname IN professor.p_firstname%type, v_lname IN  
professor.p_lastname%type)
```

```
is
```

```
begin
```

```
insert into professor values(v_profid, v_dept, v_fname,  
v_lname);
```

```
dbms_output.put_line('Record inserted');
```

```
end insert_professor;
```

```
create or replace procedure insert_project(v_projid In  
project.project_id%type, v_pname IN project.proj_name%type,  
v_pid IN project.prof_id%type, v_type IN char, v_teamname IN  
project.team_name%type, v_startdate IN  
project.start_date%type, v_duration IN  
project.duration_in_months%type, v_status IN  
project.status%type)
```

```
is
```

```
begin
```



```
insert into project values(v_projid,v_pname,v_pid, v_type,  
v_teamname, v_startdate, v_duration, v_status);
```

```
dbms_output.put_line('Record inserted');
```

```
end insert_project;
```

```
create or replace procedure insert_team(v_teamname IN  
team.team_name%type, v_projname IN  
team.proj_name%type, v_noofstudents IN  
team.no_of_students%type)
```

```
is
```

```
begin
```

```
insert into team values(v_teamname, v_projname,  
v_noofstudents);
```

```
dbms_output.put_line('Record inserted');
```

```
end insert_team;
```

SQL Worksheet

Clear Find Actions Save

```
125 insert into student values(v_rollno,v_branch,v_gradyr, v_curryr, v_projid, v_lname, v_fname);
126 end insert_student;
127
128
129 create or replace procedure insert_professor(v_profid IN professor.prof_id%type, v_dept IN professor.dept%type, v_fname IN professor.p_firstname%type, v_lname IN professor.p_lastname%type)
130 is
131 begin
132 insert into professor values(v_profid, v_dept, v_fname, v_lname);
133 dbms_output.put_line('Record inserted');
134 end insert_professor;
135
136 create or replace procedure insert_project(v_projid In project.project_id%type, v_pname IN project.proj_name%type, v_pid IN project.prof_id%type, v_type IN char, v_teamname IN project.team_name%type)
137 is
138 begin
139 insert into project values(v_profid,v_pname,v_pid, v_type, v_teamname, v_startdate, v_duration, v_status);
140
```

Procedure created.

desc Project

desc Team

desc Professor

desc STudent

desc s_specialization

desc p_specialization

select *from project;

select *from professor;

```
select *from student;
```

```
select *from team;
```

```
select *from p_specialization;
```

```
select *from s_specialization;
```

CURSORS

```
declare
```

```
v_count number(3):=0;
```

```
v_rollno s_specialization.roll_no%type;
```

```
cursor spn is
```

```
select roll_no,specialization from s_specialization where  
roll_no=v_rollno;
```

```
rec spn%rowtype;
```

```
begin
```

```
open spn;  
  
dbms_output.put_line('roll no'||chr(9)||'specialization');  
  
loop  
  
fetch spn into rec;  
  
dbms_output.put_line(rec.roll_no|| chr(9)||rec.specialization);  
  
v_count:=v_count+1;  
  
exit when spn%NOTFOUND;  
  
end loop;  
  
dbms_output.put_line('data fetched');  
  
close spn;  
  
end;
```

```
SQL Worksheet
Clear Find Actions Save Run

153 v_rollno s_specialization.roll_no%type :=102007584;
154 cursor spn is
155 select roll_no,specialization from s_specialization where roll_no=v_rollno;
156 rec spn%rowtype;
157 begin
158 open spn;
159 dbms_output.put_line('roll no'||chr(9)||'specialization');
160 loop
161 fetch spn into rec;
162 dbms_output.put_line(rec.roll_no|| chr(9)||rec.specialization);
163 v_count:=v_count+1;
164 exit when spn%NOTFOUND;
165 end loop;
166 dbms_output.put_line('data fetched');
167
168
```

Statement processed.
roll no specialization
102007584 FULL_STACK
102007584 AI
102007584 AI
data fetched

DECLARE

--x P_specialization.specialization%type := 'Dev-ops';

CURSOR find_proff IS

SELECT prof_id, specialization FROM P_specialization

WHERE specialization like '%Computer_Vision%';

ar find_proff%rowtype;

BEGIN

open find_proff;

LOOP

fetch find_proff into ar;

DBMS_OUTPUT.PUT_LINE(ar.prof_id||chr(9)||ar.specialization);

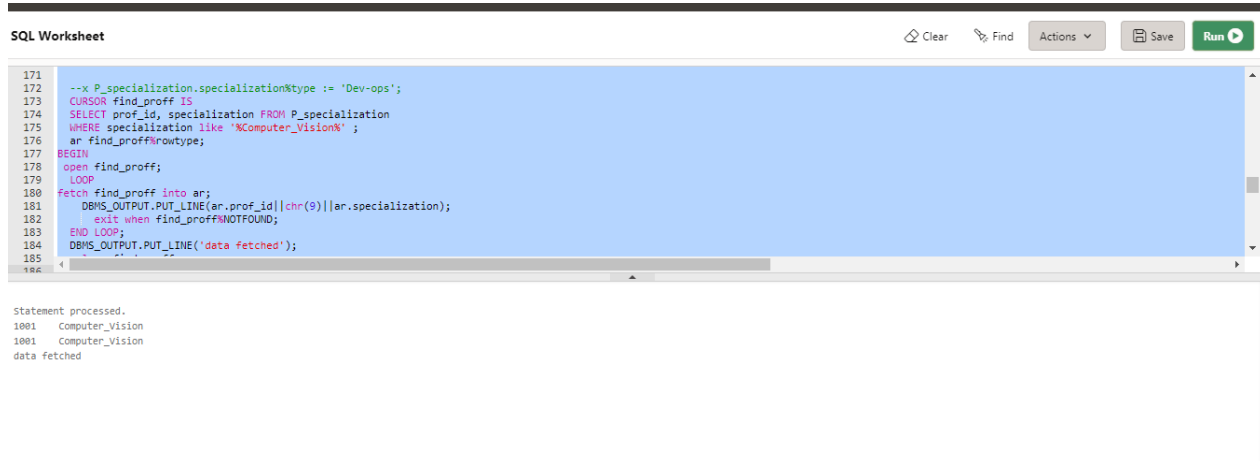
exit when find_proff%NOTFOUND;

END LOOP;

```
DBMS_OUTPUT.PUT_LINE('data fetched');

close find_proff;

END;
```



The screenshot shows an SQL Worksheet interface. The top bar includes a 'Clear' button, a 'Find' button, an 'Actions' dropdown, a 'Save' button, and a 'Run' button. The main area contains a PL/SQL script with line numbers 171 to 186. The script defines a cursor 'find_proff' and a loop that fetches data from 'P_specialization' and prints it using 'DBMS_OUTPUT.PUT_LINE'. Below the script, the output shows the statement was processed successfully and the data 'Computer_Vision' was fetched.

```
171 --x P_specialization.specialization%type := 'Dev-ops';
172 CURSOR find_proff IS
173 SELECT prof_id, specialization FROM P_specialization
174 WHERE specialization like '%Computer_Vision%';
175 an find_proff%rowtype;
176 BEGIN
177 open find_proff;
178 LOOP
179 fetch find_proff into ar;
180 DBMS_OUTPUT.PUT_LINE(ar.prof_id||chr(9)||ar.specialization);
181 exit when find_proff%NOTFOUND;
182 END LOOP;
183 DBMS_OUTPUT.PUT_LINE('data fetched');
184
185
186
```

Statement processed.
1001 Computer_Vision
1001 Computer_Vision
data fetched

FUNCTIONS

create or replace function total_students(v_branch IN
student.branch%type)

return number

is

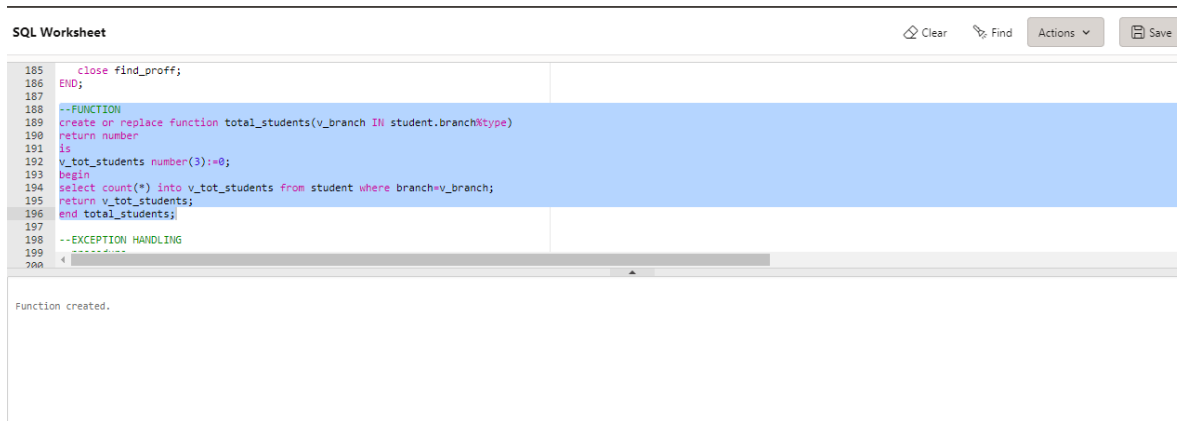
v_tot_students number(3):=0;

begin

select count(*) into v_tot_students from student where
branch=v_branch;

return v_tot_students;

end total_students;



The screenshot shows an SQL Worksheet interface with a toolbar at the top containing 'Clear', 'Find', 'Actions', and 'Save' buttons. The main area displays a PL/SQL function definition for 'total_students' starting at line 185. The function is defined as follows:

```
185 close find_proff;  
186 END;  
187  
188 --FUNCTION  
189 create or replace function total_students(v_branch IN student.branch%type)  
190 return number  
191 is  
192 v_tot_students number(3):=0;  
193 begin  
194 select count(*) into v_tot_students from student where branch=v_branch;  
195 return v_tot_students;  
196 end total_students;  
197  
198 --EXCEPTION HANDLING  
199  
200
```

Below the code editor, the output area shows the message 'Function created.'

EXCEPTION HANDLING-

--procedure

CREATE OR REPLACE PROCEDURE

INSERT_STUDENT(v_rollno IN student.Roll_no%type, v_fname

```
IN student.s_firstname%type, v_lname IN  
student.s_lastname%type, v_branch IN student.branch%type,  
v_gradyr IN student.grad_year%type, v_curryr IN  
student.curr_year%type, v_projid IN student.project_id%type)
```

```
is
```

```
begin
```

```
insert into student values(v_rollno,v_branch,v_gradyr, v_curryr,  
v_projid, v_lname, v_fname);
```

```
end insert_student;
```

```
--exception handling
```

```
DECLARE
```

```
    v_rollno student.Roll_no%type;
```

```
    -- v_fname student.s_firstname%type;
```

```
--    v_lname student.s_lastname%type;
```

```
    v_branch student.branch%type;
```

```
    -- v_gradyr student.grad_year%type;
```



```

-- v_curryr student.curr_year%type;

-- v_projid student.project_id%type;


BEGIN

    SELECT Roll_no, branch INTO v_rollno, v_branch

    FROM student where s_firstname LIKE '%TANISHA%';


    DBMS_OUTPUT.PUT_LINE ('Roll_no: ' || v_rollno);

    DBMS_OUTPUT.PUT_LINE ('Branch: ' || v_branch);

    -- WHERE s_firstname = v_fname ;

    -- WHERE s_lastname = v_lname ;

    -- WHERE grad_year = v_gradyr ;

    -- WHERE curr_year = v_curryr ;

    -- WHERE project_id = v_projid ;


EXCEPTION

```

WHEN no_data_found THEN

dbms_output.put_line('No such student!');

WHEN too_many_rows THEN

dbms_output.put_line('Many rows found') ;

WHEN others THEN

dbms_output.put_line('Error!');

END;

The screenshot shows an SQL Worksheet interface. The top bar includes 'Clear', 'Find', 'Actions', and 'Save' buttons. The main area contains a PL/SQL block with the following code:

```
221 DBMS_OUTPUT.PUT_LINE ('Branch: ' || v_branch);
222 -- WHERE s_firstname = v_fname ;
223 -- WHERE s_lastname = v_lname ;
224 -- WHERE grad_year = v_gradyr ;
225 -- WHERE curr_year = v_curryr ;
226 -- WHERE project_id = v_projid ;
227 EXCEPTION
228 WHEN no_data_found THEN
229     dbms_output.put_line('No such student!');
230 WHEN too_many_rows THEN
231     dbms_output.put_line('Many rows found') ;
232 WHEN others THEN
233     dbms_output.put_line('Error!');
234 END;
235
236
```

Below the code editor, the output area shows the result of the statement execution:

```
Statement processed.
No such student!
```

TRIGGERS-

--trigger that one project duration cannot be more than 12 months

```
create or replace trigger check_duration
before
insert of duration_in_months
on project
for each row
begin
if: duration_in_months>6 then

raise_application_error('Duration should not be more than 12
months');

end if;

end;
```

2.

--delete one project from table if status is in progress then u
cant delete

```
CREATE OR REPLACE TRIGGER delproject
```

AFTER DELETE ON project

FOR EACH ROW

DECLARE

--sal_diff number;

BEGIN

if: old.status='in progress' then
dbms_output.put_line('cannot delete project that is not
finished');

end if;

END;

--CALL TRIGGER

DECLARE v_projid project.project_id%type :=3003;

BEGIN

DELETE FROM project

WHERE project_id =v_projid;

IF sql%notfound THEN

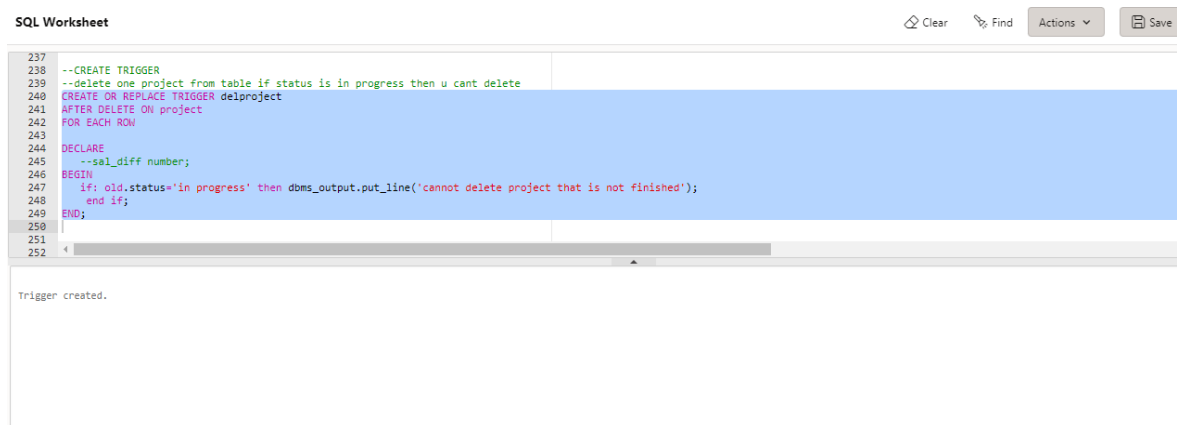
dbms_output.put_line('no project deleted');

ELSif sql%found THEN

dbms_output.put_line('one row deleted');

END IF;

END;



The screenshot shows an SQL Worksheet interface. At the top, there's a title bar with 'SQL Worksheet' on the left and 'Clear', 'Find', 'Actions', and 'Save' buttons on the right. Below the title bar is a text area containing an SQL script. The script is as follows:

```
237
238 --CREATE TRIGGER
239 --delete one project from table if status is in progress then u cant delete
240 CREATE OR REPLACE TRIGGER delproject
241 AFTER DELETE ON project
242 FOR EACH ROW
243
244 DECLARE
245     --sal_diff number;
246 BEGIN
247     if: old.status='in progress' then dbms_output.put_line('cannot delete project that is not finished');
248     end if;
249 END;
250
251
252
```

Below the text area, there's a status bar that says 'Trigger created.'

SUBMITTED BY-
GROUP 1 OF 2CO18
BE COE
TIET

Signature of Faculty member