American International University-Bangladesh (AIUB)

ADVANCE DATABASE MANAGEMENT SYSTEM

SECTION: B

GROUP NAME: AVENGERS

PROJECT TITLE: CRICKET CLUB MANAGEMENT SYSTEM

Submitted by:

SL	NAME	ID
01	HASAN, SHADIK	19-39357-1
02	HOSSEN, DIDAR	18-39084-3
03	PARVEEN, SUMAIYA	19-41542-3
04	RAHMAN, RAHAT	20-43401-1

Submitted to:

REZWAN AHMED

Table of Contents

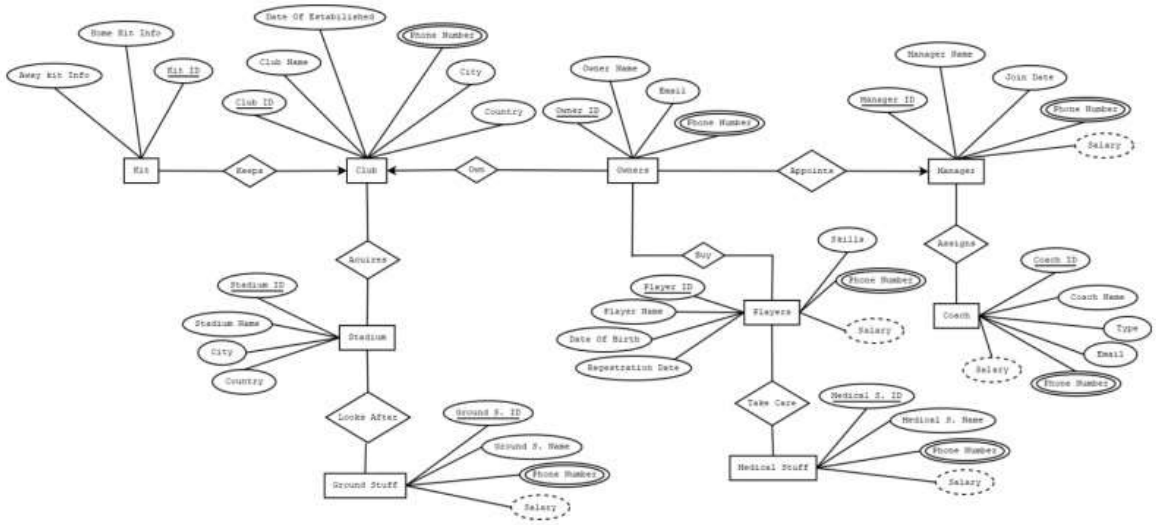
INTRODUCTION:

The title of the project is “Cricket club management system”. Cricket club management project is designed with the motive of managing a cricket club. This software system consists of various online booking and management functionaries needed by a cricket club. This software system assures efficient management and maintains the functioning on a cricket club. The main aim of cricket club management is to manage the details of cricket, players, events, fees, schedules etc. . We do believe that this project will be beneficial to a lot of new or updatable Cricket club management system.

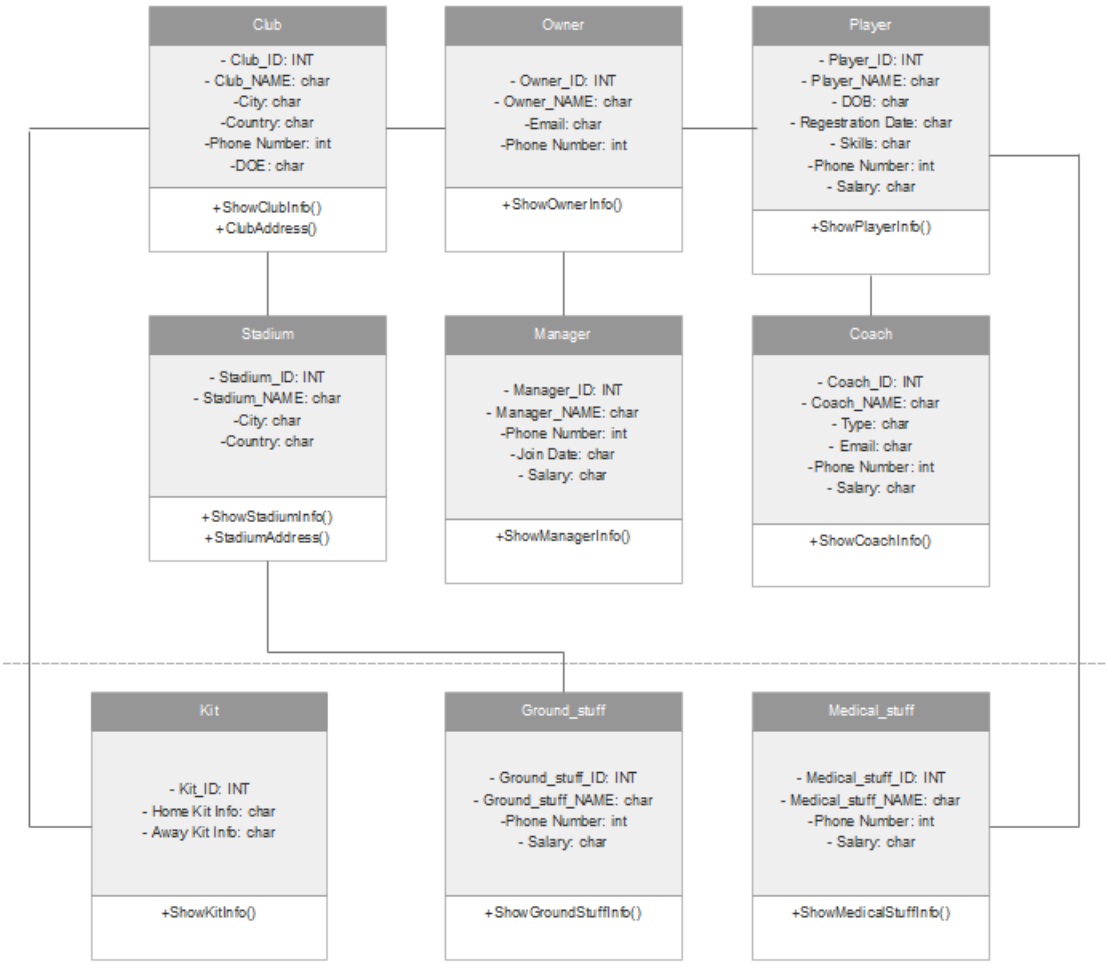
SYSTEM SUMMARY:

In Cricket Club Management System To identify a club the system stores club name, club ID, established date, location and phone numbers. Every Club acquires one or many stadiums. Every stadium has stadium name, location and one stadium ID. Ground stuffs look after those stadiums. One ground stuff has only one Id but may have more than one phone number. Club may have many Owners. A owner is identified by Owner_Id. An Owner information composed of name, email and one or more phone number. Owners appoints only one Manager for the club. Owners can buy many players. Every Player can be identified by their Player ID. Club also store players data like Player name, date of birth, registration date, skills, phone numbers and salary. Manager can assign coaches. To take care of the players club assigns medical stuffs. A medical Stuff have one medical_s.ID. The system also stores their name and phone numbers. Every coach has their individual one Id and may have more than one phone numbers. The System also store their name, address, Email address. Club keeps their Kits. kit information consists of home kit info, away kit info and one kit ID.

ER DIAGRAM:

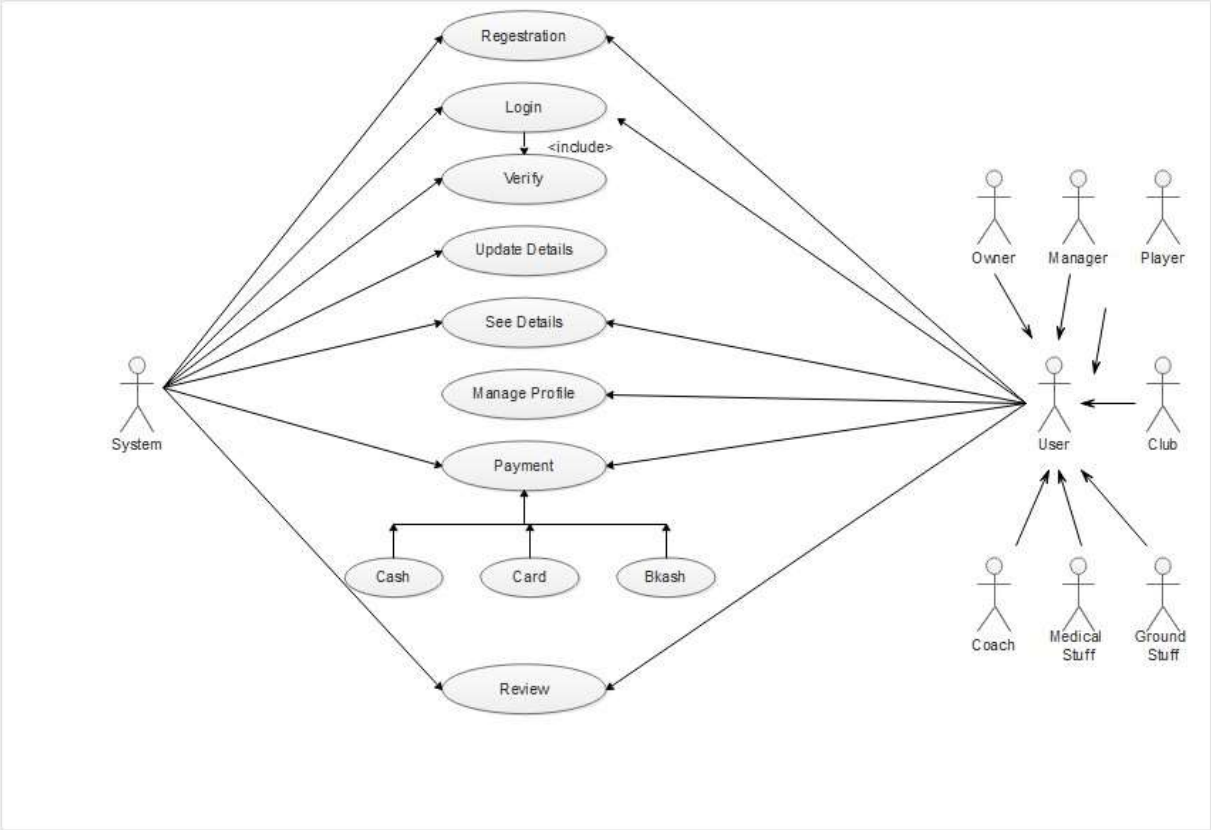

Home Kit Info
Away Kit Info
Kit ID
Kit
Keeps
Date Of Established
Phone Number
Club Name
City
Club ID
Country
Club
Owns
Owner Name
Email
Owner ID
Phone Number
Owners
Appoints
Manager Name
Join Date
Manager ID
Phone Number
Salary
Manager
Acquires
Stadium ID
Stadium Name
City
Country
Stadium
Looks After
Ground S. ID
Ground S. Name
Phone Number
Salary
Ground Stuff
Buy
Skills
Phone Number
Salary
Player ID
Player Name
Date Of Birth
Registration Date
Players
Assigns
Coach ID
Coach Name
Type
Email
Phone Number
Salary
Coach
Take Care
Medical S. ID
Medical S. Name
Phone Number
Salary
Medical Stuff

CLASS DIAGRAM:

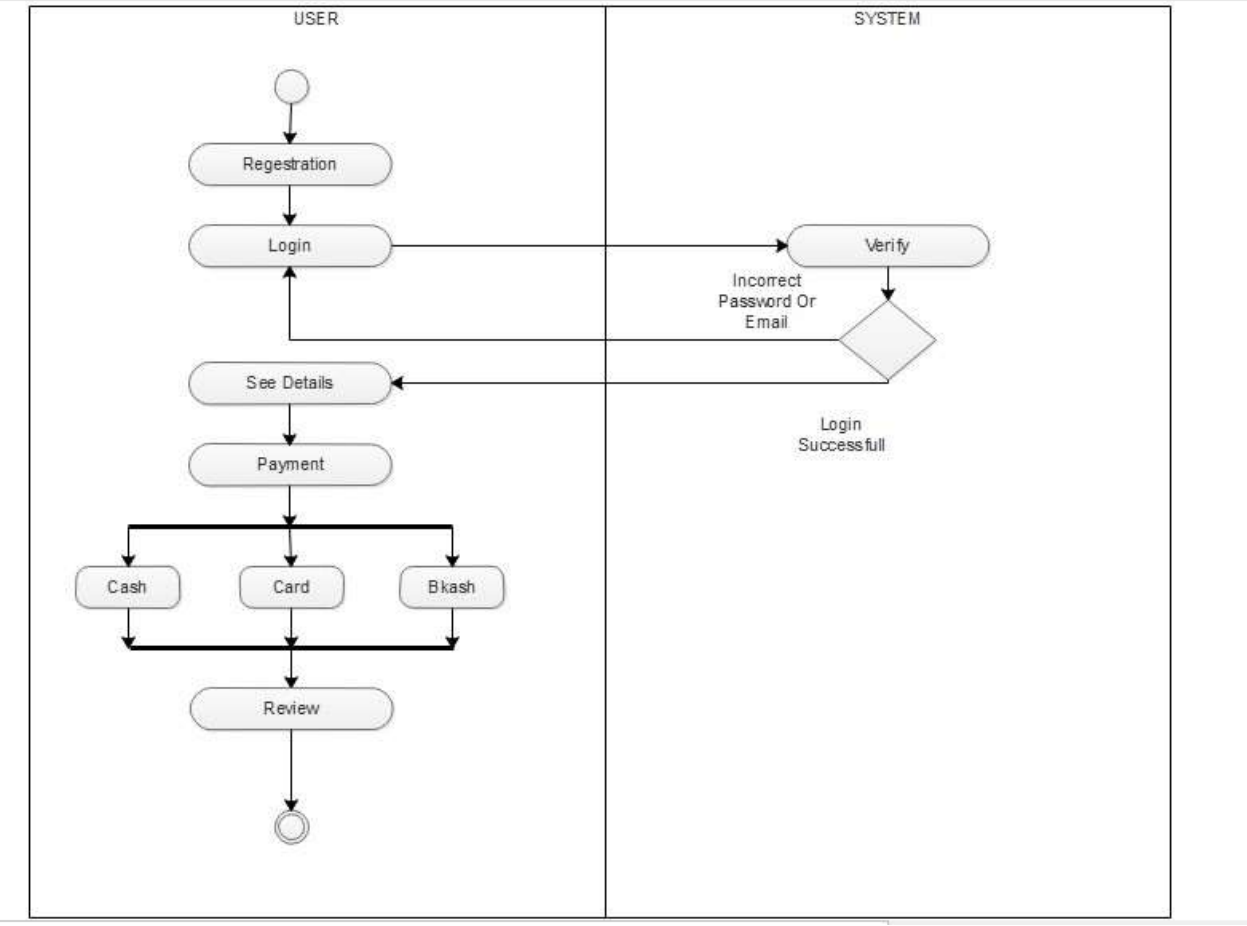


USE CASE AND ACTIVITY DIAGRAM:

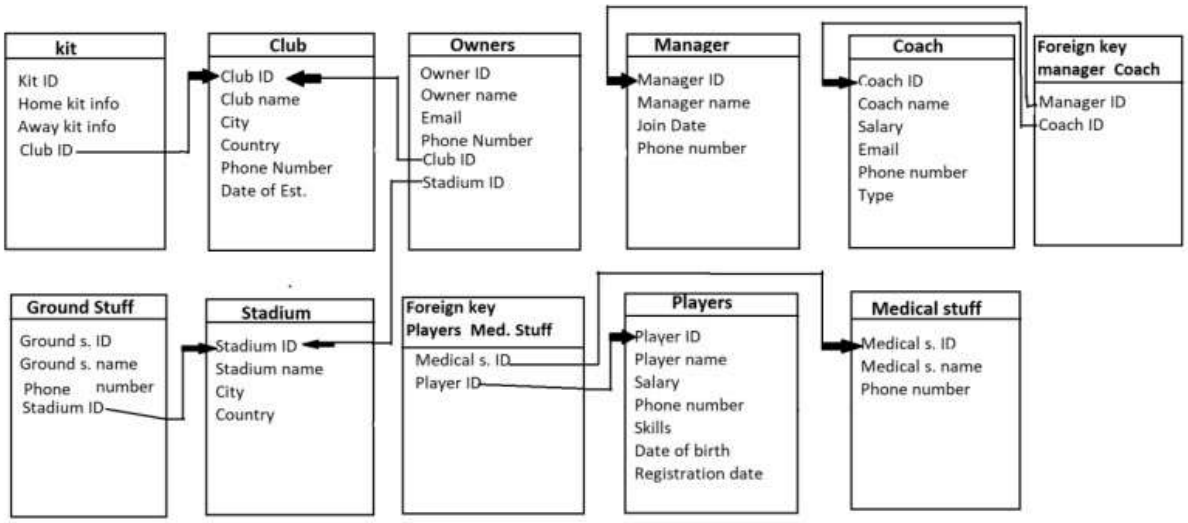
USE CASE:


Regestration
Login
<include>
Verify
Update Details
See Details
Manage Profile
Payment
Cash
Card
Bkash
Review
System
Owner
Manager
Player
User
Club
Coach
Medical Stuff
Ground Stuff

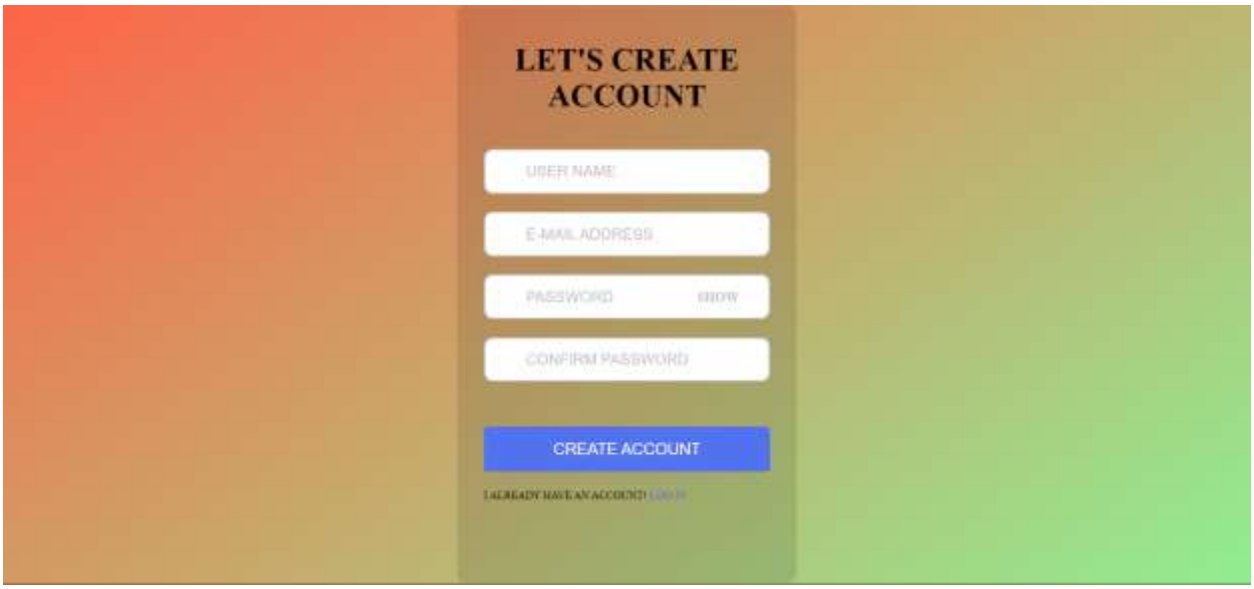
ACTIVITY:



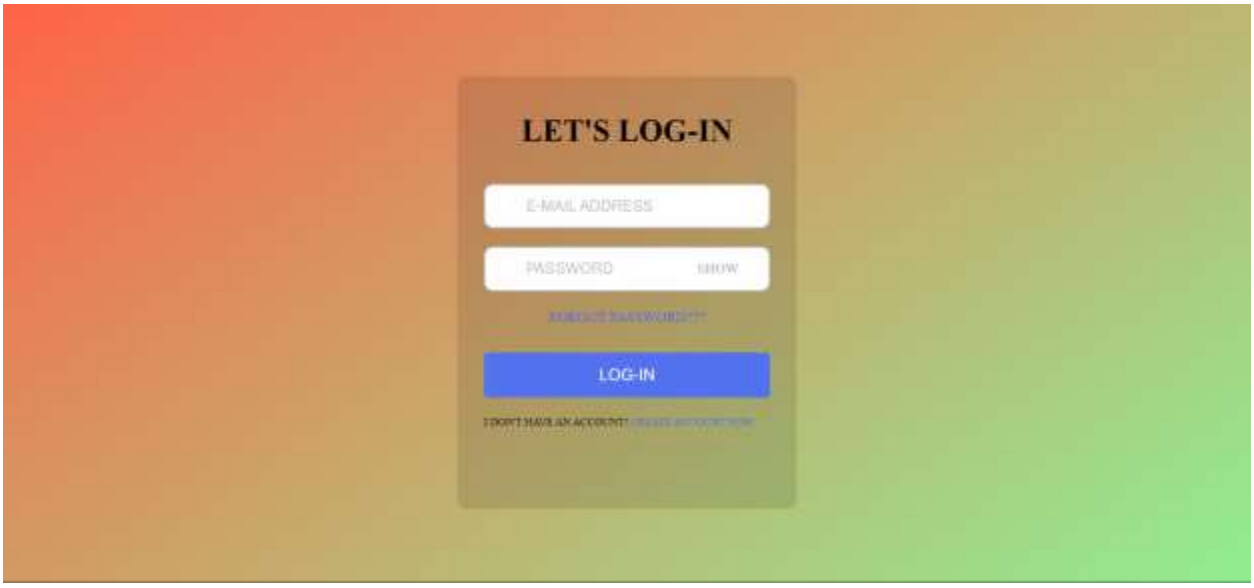
SCHEMA DIAGRAM:



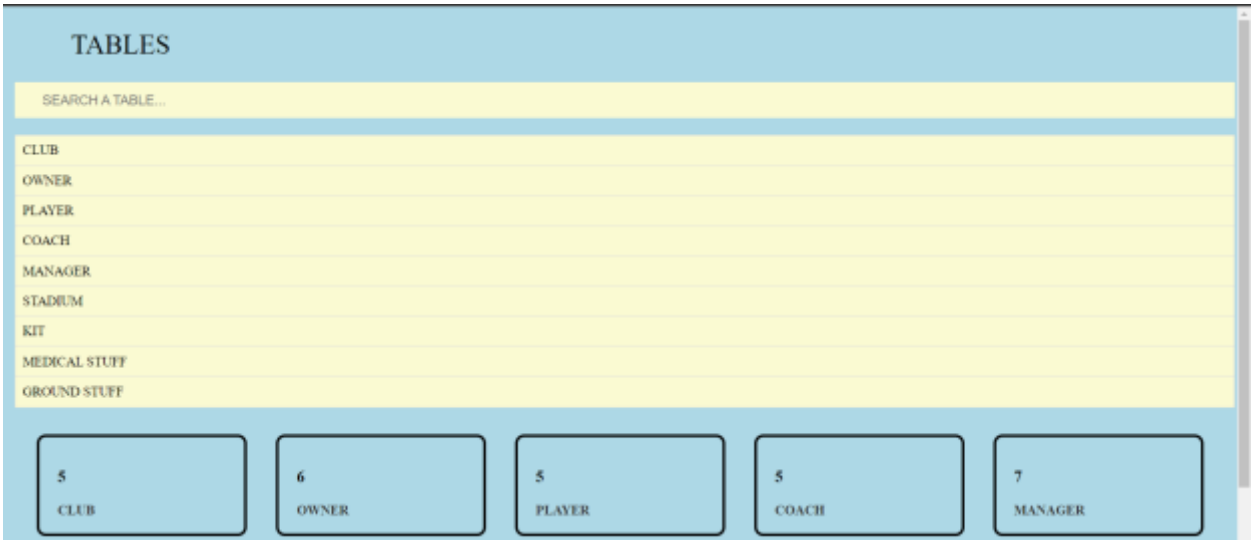
ADMIN PANEL:



This is the registration page with validation and connected with oracle 10g.



This is the Login page with validation and connected with oracle 10g.



This is a admin dashboard page. This is also connected with oracle 10g. Here the admin can search for a table and view the table page. And for the table, a count was shown that is done by the counting from oracle.

This is the page for a table and here admin can see the table which is on oracle. And also here admin can edit data for update and delete. Also if the admin can search for data he/she can search by id.

TABLE CREATION :

FOR CLUB :

CREATE TABLE CLUB (CLUB_ID NUMBER(10) PRIMARY KEY, CLUB_NAME VARCHAR(25), DATE_OF_ESTABILISHED VARCHAR(10), CLUB_PHONE_NUMBER_1 NUMBER(11), CLUB_PHONE_NUMBER_2 NUMBER(11), CLUB_PHONE_NUMBER_3 NUMBER(11));

CREATE SEQUENCE CLUB_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC CLUB;

FOR CLUB ADDRESS:

CREATE TABLE CLUB_ADDRESS (A_ID NUMBER(10) PRIMARY KEY, CLUB_CITY VARCHAR(30), CLUB_COUNTRY VARCHAR(30));

CREATE SEQUENCE A_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC CLUB;

FOR OWNER:

CREATE TABLE OWNER (OWNER_ID NUMBER(10) PRIMARY KEY, OWNER_NAME VARCHAR(30), EMAIL VARCHAR(30), OWNER_PHONE_NUMBER_1 NUMBER(11), OWNER_PHONE_NUMBER_2 NUMBER(11), OWNER_PHONE_NUMBER_3 NUMBER(11), CLUB_ID NUMBER(10), A_ID NUMBER(10), S_ID NUMBER(10), STADIUM_ID NUMBER(10));

ALTER TABLE OWNER ADD CONSTRAINT FK FOREIGN KEY (A_ID) REFERENCES CLUB_ADDRESS (A_ID);

ALTER TABLE OWNER ADD CONSTRAINT FK FOREIGN KEY (S_ID) REFERENCES STADIUM_ADDRESS (S_ID);

CREATE SEQUENCE OWNER_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC OWNER;

FOR MANAGER:

CREATE TABLE MANAGER (MANAGER_ID NUMBER(10) PRIMARY KEY, MANAGER_NAME VARCHAR(30), JOIN_DATE VARCHAR(10), MANAGER_PHONE_NUMBER_1 NUMBER(11), MANAGER_PHONE_NUMBER_2 NUMBER(11), MANAGER_PHONE_NUMBER_3 NUMBER(11), MANAGER_SALARY NUMBER(20));

CREATE SEQUENCE MANAGER_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC MANAGER;

FOR KIT:

CREATE TABLE KIT (KIT_ID NUMBER(10) PRIMARY KEY, HOME_KIT_INFO VARCHAR(30), AWAY_KIT_INFO VARCHAR(30), CLUB_ID NUMBER (10), A_ID NUMBER(10));

CREATE SEQUENCE KIT_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC KIT;

FOR PLAYER:

CREATE TABLE PLAYER (PLAYER_ID NUMBER(10) PRIMARY KEY, PLAYER_NAME VARCHAR(30), DATE_OF_BIRTH VARCHAR(10), REGESTRATION_DATE VARCHAR(10),SKILLS VARCHAR(10),PLAYER_PHONE_NUMBER_1 NUMBER(11),PLAYER_PHONE_NUMBER_2 NUMBER(11), PLAYER_PHONE_NUMBER_3 NUMBER(11), PLAYER_SALARY NUMBER(20));

CREATE SEQUENCE PLAYER_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC PLAYER;

FOR MEDICAL STUFF:

CREATE TABLE MEDICAL_STUFF (MEDICAL_S_ID NUMBER(10) PRIMARY KEY, MEDICAL_S_NAME VARCHAR(30), MEDICAL_S_PHONE_NUMBER_1 NUMBER(11), MEDICAL_S_PHONE_NUMBER_2 NUMBER(11), MEDICAL_S_PHONE_NUMBER_3 NUMBER(11), MEDICAL_S_SALARY NUMBER(20));

CREATE SEQUENCE MEDICAL_S_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC MEDICAL_STUFF;

FOR STADIUM:

CREATE TABLE STADIUM (STADIUM_ID NUMBER(10) PRIMARY KEY, STADIUM_NAME VARCHAR(30));

CREATE SEQUENCE STADIUM_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC STADIUM;

FOR STADIUM ADDRESS:

CREATE TABLE STADIUM_ADDRESS (S_ID NUMBER(10) PRIMARY KEY, STADIUM_CITY VARCHAR(30), STADIUM_COUNTRY VARCHAR(30));

CREATE SEQUENCE S_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC STADIUM_ADDRESS;

FOR GROUND STUFF:

CREATE TABLE GROUND_STUFF (GROUND_S_ID NUMBER(10) PRIMARY KEY, GROUND_S_NAME VARCHAR(30), GROUND_S_PHONE_NUMBER_1 NUMBER(11), GROUND_S_PHONE_NUMBER_2 NUMBER(11), GROUND_S_PHONE_NUMBER_3 NUMBER(11), GROUND_S_SALARY NUMBER(20), S_ID NUMBER(10), STADIUM_ID NUMBER(10));

ALTER TABLE GROUND_STUFF ADD CONSTRAINT GK FOREIGN KEY (S_ID) REFERENCES STADIUM_ADDRESS (S_ID);

CREATE SEQUENCE GROUND_S_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC GROUND_STUFF;

FOR COACH:

CREATE TABLE COACH (COACH_ID NUMBER(10) PRIMARY KEY, COACH_NAME VARCHAR(30),TYPE VARCHAR(30), EMAIL VARCHAR(30), COACH_PHONE_NUMBER_1 NUMBER(11), COACH_PHONE_NUMBER_2 NUMBER(11), COACH_PHONE_NUMBER_3 NUMBER(11), COACH_SALARY NUMBER(20));

CREATE SEQUENCE COACH_ID_SEQ START WITH 1001 MAXVALUE 90000 INCREMENT BY 50 NOCACHE NOCYCLE;

DESC COACH;

FOR FOREIGN KEY OWNER PLAYER:

CREATE TABLE FOREIGN_KEY_OWNER_PLAYER (OWNER_ID NUMBER(10), PLAYER_ID NUMBER(10));

ALTER TABLE FOREIGN_KEY_OWNER_PLAYER ADD CONSTRAINT IF FOREIGN KEY (OWNER_ID) REFERENCES OWNER (OWNER_ID);

ALTER TABLE FOREIGN_KEY_OWNER_PLAYER ADD CONSTRAINT LF FOREIGN KEY (PLAYER_ID) REFERENCES PLAYER (PLAYER_ID);

DESC FOREIGN_KEY_OWNER_PLAYER;

FOR FOREIGN KEY PLAYER MEDICAL STUFF:

CREATE TABLE FOREIGN_KEY_PLAYER_MS (PLAYER_ID NUMBER(10), MEDICAL_S_ID NUMBER(10));

ALTER TABLE FOREIGN_KEY_PLAYER_MS ADD CONSTRAINT MF FOREIGN KEY (PLAYER_ID) REFERENCES PLAYER (PLAYER_ID);

ALTER TABLE FOREIGN_KEY_PLAYER_MS ADD CONSTRAINT AF FOREIGN KEY (MEDICAL_S_ID) REFERENCES MEDICAL_STUFF (MEDICAL_S_ID);

DESC FOREIGN_KEY_PLAYER_MS;

FOR FOREIGN KEY OWNER STADIUM:

CREATE TABLE FOREIGN_KEY_OWNER_STADIUM (OWNER_ID NUMBER(10), STADIUM_ID NUMBER(10));

ALTER TABLE FOREIGN_KEY_OWNER_STADIUM ADD CONSTRAINT HF FOREIGN KEY (OWNER_ID) REFERENCES OWNER (OWNER_ID);

ALTER TABLE FOREIGN_KEY_OWNER_STADIUM ADD CONSTRAINT BF FOREIGN KEY (STADIUM_ID) REFERENCES STADIUM (STADIUM_ID);

DESC FOREIGN_KEY_OWNER_STADIUM;

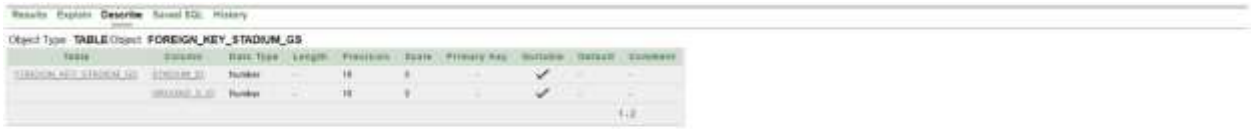
FOR FOREIGN KEY STADIUM GROUND STUFF:

CREATE TABLE FOREIGN_KEY_STADIUM_GS (STADIUM_ID NUMBER(10), GROUND_S_ID NUMBER(10));

ALTER TABLE FOREIGN_KEY_STADIUM_GS ADD CONSTRAINT UF FOREIGN KEY (STADIUM_ID) REFERENCES STADIUM (STADIUM_ID);

ALTER TABLE FOREIGN_KEY_STADIUM_GS ADD CONSTRAINT VF FOREIGN KEY (GROUND_S_ID) REFERENCES GROUND_STUFF (GROUND_S_ID);

DESC FOREIGN_KEY_STADIUM_GS;



FOR FOREIGN KEY MANAGER COACH:

CREATE TABLE FOREIGN_KEY_MANAGER_COACH (MANAGER_ID NUMBER(10), COACH_ID NUMBER(10));

ALTER TABLE FOREIGN_KEY_MANAGER_COACH ADD CONSTRAINT CF FOREIGN KEY (MANAGER_ID) REFERENCES MANAGER (MANAGER_ID);

ALTER TABLE FOREIGN_KEY_MANAGER_COACH ADD CONSTRAINT DF FOREIGN KEY (COACH_ID) REFERENCES COACH (COACH_ID);

DESC FOREIGN_KEY_MANAGER_COACH;

DATA INSERTION:

FOR CLUB :

INSERT INTO CLUB VALUES (CLUB_ID_SEQ.NEXTVAL, 'DHAKA PLATOON', '2015', '094652641', '058412656', '0179654166');

INSERT INTO CLUB VALUES (CLUB_ID_SEQ.NEXTVAL, 'RAJSHAHI ROYELS', '2012', '09545463', '01785621265', '01941635126');

INSERT INTO CLUB VALUES (CLUB_ID_SEQ.NEXTVAL, 'RANGPUR RANGERS', '2012', '054308523', '0241574521', '01815461265');

INSERT INTO CLUB VALUES (CLUB_ID_SEQ.NEXTVAL, 'CUMILLA WARRIORS', '2015', '0841651', '01485126352', '0135421851');

INSERT INTO CLUB VALUES (CLUB_ID_SEQ.NEXTVAL, 'BARISHAL BULLS', '2012', '87486542', '9886464', '01796554486');

SELECT * FROM CLUB;

FOR CLUB ADDRESS:

INSERT INTO CLUB_ADDRESS VALUES (A_ID_SEQ.NEXTVAL, 'DHAKA', 'BANGLADESH');

INSERT INTO CLUB_ADDRESS VALUES (A_ID_SEQ.NEXTVAL, 'RAJSHAHI', 'BANGLADESH');

INSERT INTO CLUB_ADDRESS VALUES (A_ID_SEQ.NEXTVAL, 'RANGPUR', 'BANGLADESH');

INSERT INTO CLUB_ADDRESS VALUES (A_ID_SEQ.NEXTVAL, 'CUMILLA', 'BANGLADESH');

INSERT INTO CLUB_ADDRESS VALUES (A_ID_SEQ.NEXTVAL, 'BARISHAL', 'BANGLADESH');

SELECT * FROM CLUB_ADDRESS;

Results Explain Describe Saved SQL History

A_ID	CLUB_CITY	CLUB_COUNTRY
1001	DHAKA	BANGLADESH
1051	RAJSHAHI	BANGLADESH
1101	RANGPUR	BANGLADESH
1151	CUMILLA	BANGLADESH
1201	BARISHAL	BANGLADESH

5 rows returned in 0.00 seconds CSV Export

FOR OWNER:

INSERT INTO OWNER VALUES (OWNER_ID_SEQ.NEXTVAL, 'BEXIMCO GROUP', 'beximco@gmail.com', '065416953', '06526316', '01865326541', '1001', '1001', '1151', '1201');

INSERT INTO OWNER VALUES (OWNER_ID_SEQ.NEXTVAL, 'BANGLADESH CRICKET BOARD', 'bcb@gmail.com', '0695165', '09654152', '036525463', '1051', '1051', '1201', '1251');

INSERT INTO OWNER VALUES (OWNER_ID_SEQ.NEXTVAL, 'BANGLADESH CRICKET BOARD', 'bcb@gmail.com', '84541206321', '5841665', '014584169', '1201', '1101', '1251', '1301');

INSERT INTO OWNER VALUES (OWNER_ID_SEQ.NEXTVAL, 'MUSTAFA RAFIQUL ISLAM', 'mustafa@gmail.com', '01986526', '01965219', '0106526341', '1251', '1101', '1301', '1351');

INSERT INTO OWNER VALUES (OWNER_ID_SEQ.NEXTVAL, 'NAFISA KAMAL', 'nafisa@gmail.com', '08562652', '048541652', '05849652', '1301', '1201', '1351', '1401');

SELECT * FROM OWNER;

FOR MANAGER:

INSERT INTO MANAGER VALUES (MANAGER_ID_SEQ.NEXTVAL, 'FARHAN', '15-10-2017', '01085165846', '018542966', '08541965', '50000');

INSERT INTO MANAGER VALUES (MANAGER_ID_SEQ.NEXTVAL, 'FARIZ', '18-12-2019', '086529652', '0874296635', '056412965', '40000');

INSERT INTO MANAGER VALUES (MANAGER_ID_SEQ.NEXTVAL, 'SADI', '01-06-2020', '058654296', '0541896526', '01875496', '50000');

INSERT INTO MANAGER VALUES (MANAGER_ID_SEQ.NEXTVAL, 'MUSTAFA', '03-02-2018', '06526952', '056296635', '089645296', '45000');

INSERT INTO MANAGER VALUES (MANAGER_ID_SEQ.NEXTVAL, 'NAFI', '09-08-2021', '085461265', '096412966', '056429652', '50000');

SELECT * FROM MANAGER;

FOR KIT:

INSERT INTO KIT VALUES (KIT_ID_SEQ.NEXTVAL, 'YELLOW', 'GREEN', CLUB_ID_SEQ.NEXTVAL, A_ID_SEQ.NEXTVAL);

INSERT INTO KIT VALUES (KIT_ID_SEQ.NEXTVAL, 'BLUE', 'ORANGE', CLUB_ID_SEQ.NEXTVAL, A_ID_SEQ.NEXTVAL);

INSERT INTO KIT VALUES (KIT_ID_SEQ.NEXTVAL, 'BLACK', 'RED', CLUB_ID_SEQ.NEXTVAL, A_ID_SEQ.NEXTVAL);

INSERT INTO KIT VALUES (KIT_ID_SEQ.NEXTVAL, 'WHITE', 'PURPLE', CLUB_ID_SEQ.NEXTVAL, A_ID_SEQ.NEXTVAL);

INSERT INTO KIT VALUES (KIT_ID_SEQ.NEXTVAL, 'BLACK','ORANGE', CLUB_ID_SEQ.NEXTVAL, A_ID_SEQ.NEXTVAL);

SELECT * FROM KIT;

Results Explain Describe Saved SQL History

KIT_ID	HOME_KIT_INFO	AWAY_KIT_INFO	CLUB_ID	A_ID
1101	YELLOW	GREEN	1251	1251
1151	BLUE	ORANGE	1301	1301
1201	BLACK	RED	1351	1351
1251	WHITE	PURPLE	1401	1401
1301	BLACK	ORANGE	1451	1451

5 rows returned in 0.00 seconds CSV Export

FOR PLAYER:

INSERT INTO PLAYER VALUES (PLAYER_ID_SEQ.NEXTVAL, 'MASHRAFI MORTAZA', '05-10-1983', '01-09-2019', 'BOWLER', '0451856165', '0145865412', '018529652', '80000');

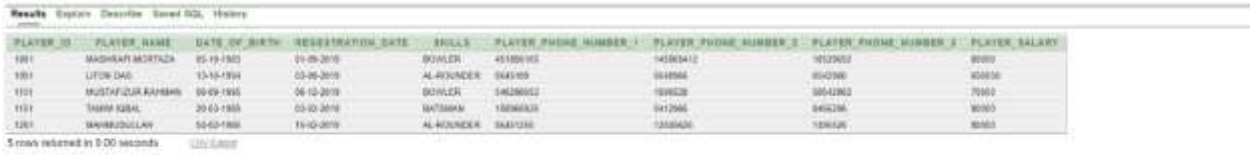
INSERT INTO PLAYER VALUES (PLAYER_ID_SEQ.NEXTVAL, 'LITON DAS', '13-10-1994', '03-08-2019', 'AL-ROUNDER','08645189', '05648966', '06542986', '65000');

INSERT INTO PLAYER VALUES (PLAYER_ID_SEQ.NEXTVAL, 'MUSTAFIZUR RAHMAN','06-09-1995', '06-12-2019', 'BOWLER', '0546298652', '01896528', '058642962', '70000');

INSERT INTO PLAYER VALUES (PLAYER_ID_SEQ.NEXTVAL, 'TAMIM IQBAL', '20-03-1989','03-02-2019', 'BATSMAN', '0188965926', '05412965', '08456296', '80000');

INSERT INTO PLAYER VALUES (PLAYER_ID_SEQ.NEXTVAL, 'MAHMUDULLAH', '04-02-1986', '15-02-2019', 'AL-ROUNDER','086451256', '012845626', '01496526', '80000');

SELECT * FROM PLAYER;



FOR MEDICAL STUFF:

INSERT INTO MEDICAL_STUFF VALUES (MEDICAL_S_ID_SEQ.NEXTVAL, 'DR. ABIR', '084758415', '0586542085', '05418566', '30000');

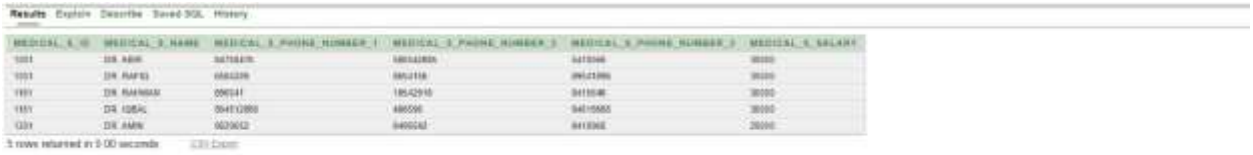
INSERT INTO MEDICAL_STUFF VALUES (MEDICAL_S_ID_SEQ.NEXTVAL, 'DR. RAFIQ', '06584289', '08654156', '089541896', '35000');

INSERT INTO MEDICAL_STUFF VALUES (MEDICAL_S_ID_SEQ.NEXTVAL, 'DR. RAHMAN', '0896541', '018542916', '08415646', '30000');

INSERT INTO MEDICAL_STUFF VALUES (MEDICAL_S_ID_SEQ.NEXTVAL, 'DR. IQBAL', '0564512856', '0486596', '094615665', '30000');

INSERT INTO MEDICAL_STUFF VALUES (MEDICAL_S_ID_SEQ.NEXTVAL, 'DR. AMIN', '09529652', '08496542', '08418965', '25000');

SELECT * FROM MEDICAL_STUFF ;



FOR STADIUM:

INSERT INTO STADIUM VALUES (STADIUM_ID_SEQ.NEXTVAL, 'MIRPUR NATIONAL STADIUM');

INSERT INTO STADIUM VALUES (STADIUM_ID_SEQ.NEXTVAL, 'SAHID A.H.M. STADIUM');

INSERT INTO STADIUM VALUES (STADIUM_ID_SEQ.NEXTVAL, 'RANGPUR STADIUM');

INSERT INTO STADIUM VALUES (STADIUM_ID_SEQ.NEXTVAL, 'BARISAL SOINIK STADIUM');

INSERT INTO STADIUM VALUES (STADIUM_ID_SEQ.NEXTVAL, 'SHADID ABDUR RAB STADIUM');

SELECT * FROM STADIUM;

Results Explain Describe Saved SQL History

STADIUM_ID	STADIUM_NAME
1001	MIRPUR NATIONAL STADIUM
1051	SAHID A.H.M. STADIUM
1101	RANGPUR STADIUM
1151	BARISAL SOINIK STADIUM
1201	SHADID ABDUR RAB STADIUM

5 rows returned in 0.00 seconds CSV Export

FOR STADIUM ADDRESS:

INSERT INTO STADIUM_ADDRESS VALUES (S_ID_SEQ.NEXTVAL, 'DHAKA', 'BANGLADESH');

INSERT INTO STADIUM_ADDRESS VALUES (S_ID_SEQ.NEXTVAL, 'RAJSHAHI', 'BANGLADESH');

INSERT INTO STADIUM_ADDRESS VALUES (S_ID_SEQ.NEXTVAL, 'RANGPUR', 'BANGLADESH');

INSERT INTO STADIUM_ADDRESS VALUES (S_ID_SEQ.NEXTVAL, 'BARISAL', 'BANGLADESH');

INSERT INTO STADIUM_ADDRESS VALUES (S_ID_SEQ.NEXTVAL, 'CUMILLA', 'BANGLADESH');

SELECT * FROM STADIUM_ADDRESS;

Results Explain Describe Saved SQL History

S_ID	STADIUM_CITY	STADIUM_COUNTRY
1001	DHAKA	BANGLADESH
1051	RAJSHAHI	BANGLADESH
1101	RANGPUR	BANGLADESH
1151	BARISAL	BANGLADESH
1201	CUMILLA	BANGLADESH

5 rows returned in 0.00 seconds CSV Export

FOR GROUND STUFF:

INSERT INTO GROUND_STUFF VALUES (GROUND_S_ID_SEQ.NEXTVAL, 'ABIR', '08956165', '07854145', '084155', '15000', '1151', '1201');

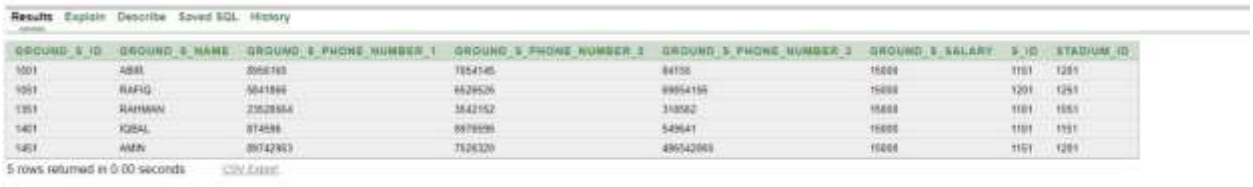
INSERT INTO GROUND_STUFF VALUES (GROUND_S_ID_SEQ.NEXTVAL, 'RAFIQ', '05841866', '06526526', '069854156', '15000', '1201', '1251');

INSERT INTO GROUND_STUFF VALUES (GROUND_S_ID_SEQ.NEXTVAL, 'RAHMAN', '023528564', '03542152', '0318562', '15000', '1101', '1051');

INSERT INTO GROUND_STUFF VALUES (GROUND_S_ID_SEQ.NEXTVAL, 'IQBAL', '0874596', '08976596', '0549641', '15000', '1101', '1151');

INSERT INTO GROUND_STUFF VALUES (GROUND_S_ID_SEQ.NEXTVAL, 'AMIN', '089742963', '07526320', '0496542065', '15000', '1151', '1201');

SELECT * FROM GROUND_STUFF ;



FOR COACH:

INSERT INTO COACH VALUES (COACH_ID_SEQ.NEXTVAL, 'KHALED MAHMUD', 'AL_ROUNDER', 'khaled@gmail.com', '0785412', '05419655', '0185496', '50000');

INSERT INTO COACH VALUES (COACH_ID_SEQ.NEXTVAL, 'OWISH SAHA', 'BATTING', 'owish@gmail.com', '08562966', '054196526', '065296352', '50000');

INSERT INTO COACH VALUES (COACH_ID_SEQ.NEXTVAL, 'MARK O DONNELL', 'BATTING', 'mark@gmail.com', '8418541', '9749652', '0564521096', '60000');

INSERT INTO COACH VALUES (COACH_ID_SEQ.NEXTVAL, 'OTTIS GIBSON', 'AL_ROUNDER', 'ottis@gmail.com', '024965269', '03251854', '085741896', '50000');

INSERT INTO COACH VALUES (COACH_ID_SEQ.NEXTVAL, 'DAV WHATMORE', 'AL_ROUNDER', 'dav@gmail.com', '08541895632', '01856296', '04896526', '60000');

SELECT * FROM COACH;

FOR FOREIGN KEY OWNER PLAYER:

INSERT INTO FOREIGN_KEY_OWNER_PLAYER VALUES ('1051', '1001');

INSERT INTO FOREIGN_KEY_OWNER_PLAYER VALUES ('1101', '1051');

INSERT INTO FOREIGN_KEY_OWNER_PLAYER VALUES ('1251', '1101');

INSERT INTO FOREIGN_KEY_OWNER_PLAYER VALUES ('1301', '1151');

INSERT INTO FOREIGN_KEY_OWNER_PLAYER VALUES ('1351', '1251');

SELECT * FROM FOREIGN_KEY_OWNER_PLAYER;

Results Explain Describe Saved SQL History

OWNER_ID	PLAYER_ID
1051	1001
1101	1051
1251	1101
1301	1151
1351	1251

5 rows returned in 0.00 seconds CSV Export

FOR FOREIGN KEY PLAYER MEDICAL STUFF:

INSERT INTO FOREIGN_KEY_PLAYER_MS VALUES ('1001', '1001');

INSERT INTO FOREIGN_KEY_PLAYER_MS VALUES ('1051', '1051');

INSERT INTO FOREIGN_KEY_PLAYER_MS VALUES ('1101', '1101');

INSERT INTO FOREIGN_KEY_PLAYER_MS VALUES ('1151', '1151');

INSERT INTO FOREIGN_KEY_PLAYER_MS VALUES ('1251', '1201');

SELECT * FROM FOREIGN_KEY_PLAYER_MS;

Results Explain Describe Saved SQL History

PLAYER_ID	MEDICAL_S_ID
1001	1001
1051	1051
1101	1101
1151	1151
1251	1201

5 rows returned in 0.02 seconds CSV Export

FOR FOREIGN KEY OWNER STADIUM:

INSERT INTO FOREIGN_KEY_OWNER_STADIUM VALUES ('1051', '1001');

INSERT INTO FOREIGN_KEY_OWNER_STADIUM VALUES ('1101', '1051');

INSERT INTO FOREIGN_KEY_OWNER_STADIUM VALUES ('1251', '1101');

INSERT INTO FOREIGN_KEY_OWNER_STADIUM VALUES ('1301', '1151');

INSERT INTO FOREIGN_KEY_OWNER_STADIUM VALUES ('1351', '1201');

SELECT * FROM FOREIGN_KEY_OWNER_STADIUM;

Results Explain Describe Saved SQL History

OWNER_ID	STADIUM_ID
1051	1001
1101	1051
1251	1101
1301	1151
1351	1201

5 rows returned in 0.00 seconds CSV Export

FOR FOREIGN KEY STADIUM GROUND STUFF:

INSERT INTO FOREIGN_KEY_STADIUM_GS VALUES ('1001', '1001');

INSERT INTO FOREIGN_KEY_STADIUM_GS VALUES ('1051', '1051');

INSERT INTO FOREIGN_KEY_STADIUM_GS VALUES ('1101', '1351');

INSERT INTO FOREIGN_KEY_STADIUM_GS VALUES ('1151', '1401');

INSERT INTO FOREIGN_KEY_STADIUM_GS VALUES ('1201', '1451');

SELECT * FROM FOREIGN_KEY_OWNER_STADIUM;

Results Explain Describe Saved SQL History

OWNER_ID	STADIUM_ID
1051	1001
1101	1051
1251	1101
1301	1151
1351	1201

5 rows returned in 0.00 seconds CSV Export

FOR FOREIGN KEY MANAGER COACH:

INSERT INTO FOREIGN_KEY_MANAGER_COACH VALUES ('1001', '1001');

INSERT INTO FOREIGN_KEY_MANAGER_COACH VALUES ('1051', '1051');

INSERT INTO FOREIGN_KEY_MANAGER_COACH VALUES ('1101', '1101');

INSERT INTO FOREIGN_KEY_MANAGER_COACH VALUES ('1151', '1151');

INSERT INTO FOREIGN_KEY_MANAGER_COACH VALUES ('1201', '1201');

SELECT * FROM FOREIGN_KEY_MANAGER_COACH;

Results Explain Describe Saved SQL History

MANAGER_ID	COACH_ID
1001	1001
1051	1051
1101	1101
1151	1151
1201	1201

5 rows returned in 0.00 seconds CSV Export

QUERY WRITING:

01. Display the player names who earn more than LITON DAS.

SELECT PLAYER_NAME FROM PLAYER WHERE PLAYER_SALARY > (SELECT PLAYER_SALARY FROM PLAYER WHERE PLAYER_NAME='LITON DAS');

Results Explain Describe Saved SQL History

PLAYER_NAME
MASHRAFI MORTAZA
MUSTAFIZUR RAHMAN
TAMIM IQBAL
MAHMUDULLAH

4 rows returned in 0.00 seconds CSV Export

02. Display the manager names who joined after FARHAN.

SELECT * FROM MANAGER WHERE JOIN_DATE < (SELECT JOIN_DATE FROM MANAGER WHERE MANAGER_NAME = 'FARHAN');

03. Display the name of owners who is from BANGLADESH.

SELECT OWNER.OWNER_NAME, STADIUM_ADDRESS.STADIUM_COUNTRY FROM OWNER, STADIUM_ADDRESS WHERE OWNER.S_ID = STADIUM_ADDRESS.S_ID AND STADIUM_ADDRESS.STADIUM_COUNTRY = 'BANGLADESH';

Results Explain Describe Saved SQL History

OWNER_NAME	STADIUM_COUNTRY
BEXIMCO GROUP	BANGLADESH
BANGLADESH CRICKET BOARD	BANGLADESH
BANGLADESH CRICKET BOARD	BANGLADESH
MUSTAFA RAFIQUL ISLAM	BANGLADESH
NAFISA KAMAL	BANGLADESH

5 rows returned in 0.00 seconds CSV Export

04. Display the home kit info of all clubs.

SELECT KIT.HOME_KIT_INFO, CLUB.CLUB_NAME FROM KIT, CLUB WHERE KIT.KIT_ID = CLUB.CLUB_ID;

Results Explain Describe Saved SQL History

HOME_KIT_INFO	CLUB_NAME
YELLOW	RANGPUR RANGERS
BLUE	CUMILLA WARRIORS
BLACK	BARISHAL BULLS

3 rows returned in 0.01 seconds CSV Export

05. Find out the skills of a player with maximum no of average salary.

SELECT SKILLS FROM PLAYER WHERE PLAYER_ID=(SELECT PLAYER_ID FROM PLAYER GROUP BY PLAYER_ID HAVING AVG(PLAYER_SALARY) = (SELECT MAX(AVG(PLAYER_SALARY))FROM PLAYER GROUP BY PLAYER_ID));

Results Explain Describe Saved SQL History

SKILLS
AL-ROUNDER

1 rows returned in 0.01 seconds CSV Export

06. Find out the coach which salary is 40000 to 50000.

SELECT * FROM COACH WHERE COACH_SALARY BETWEEN 40000 AND 50000;

Results Explain Describe Saved SQL History

COACH_ID	COACH_NAME	TYPE	EMAIL	COACH_PHONE_NUMBER_1	COACH_PHONE_NUMBER_2	COACH_PHONE_NUMBER_3	COACH_SALARY
1001	KHALED MAHMUD	AL_ROUNDER	khaled@gmail.com	785412	5419855	185496	50000
1051	OWISH SAHA	BATTING	owish@gmail.com	8562966	54196526	65296352	50000
1151	OTTIS GIBSON	AL_ROUNDER	ottis@gmail.com	24965269	3251854	85741898	50000

3 rows returned in 0.00 seconds CSV Export

07. Find the player who got second height salary

SELECT MAX(PLAYER_SALARY) FROM PLAYER WHERE PLAYER_SALARY < (SELECT MAX(PLAYER_SALARY) FROM PLAYER);

08. Find first and last record of club.

SELECT * FROM CLUB WHERE CLUB_ID = (SELECT MIN(CLUB_ID) FROM CLUB) OR CLUB_ID = (SELECT MAX(CLUB_ID) FROM CLUB);

VIEW:

01. Create a view called Club_View based on Club_Name and Club_ID from the Club Table

CREATE OR REPLACE VIEW CLUB_VIEW AS SELECT CLUB_NAME, CLUB_ID FROM CLUB;

SELECT * FROM CLUB_VIEW;

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE OR REPLACE VIEW CLUB_VIEW AS SELECT CLUB_NAME, CLUB_ID FROM CLUB;
SELECT * FROM CLUB_VIEW;
```

Results Explain Describe Saved SQL History

View created.

0.13 seconds

Results Explain Describe Saved SQL History

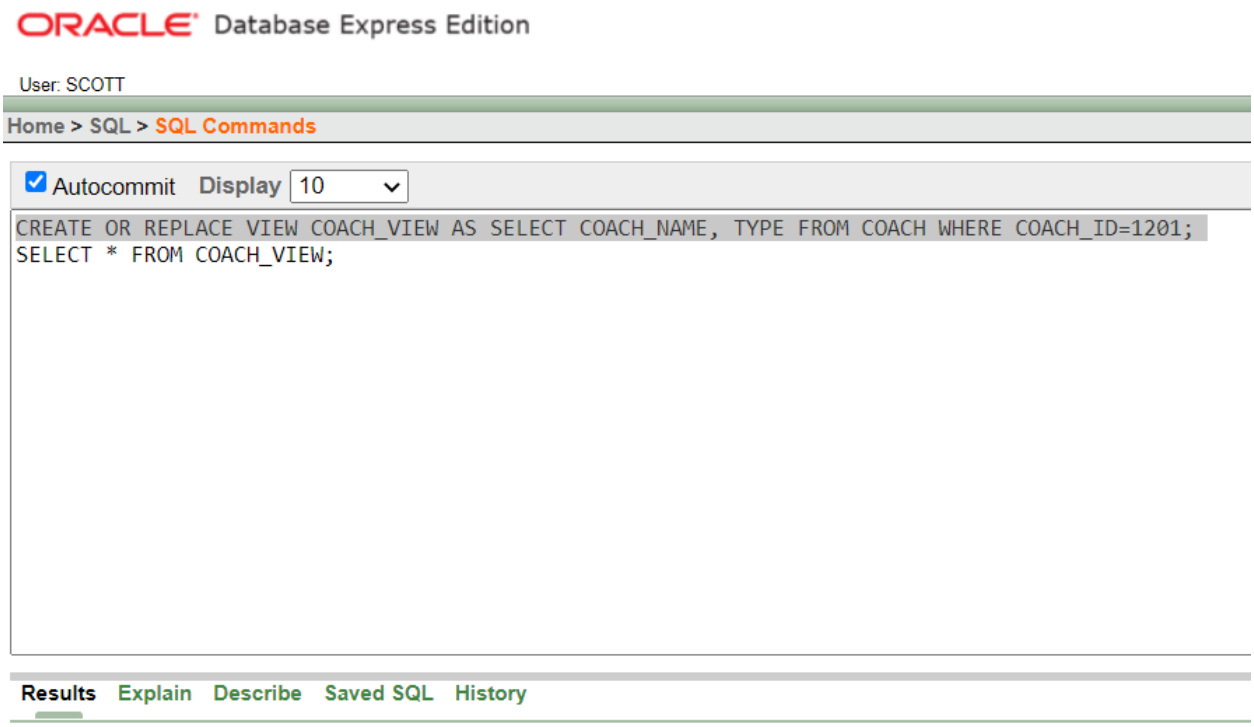
CLUB_NAME	CLUB_ID
DHAKA PLATOON	1001
RAJSHAHI ROYELS	1051
RANGPUR RANGERS	1101
CUMILLA WARRIORS	1151
BARISHAL BULLS	1201

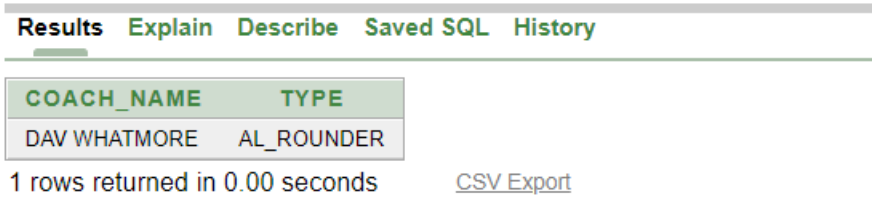
5 rows returned in 0.00 seconds CSV Export

02. Create a view to show the name and type of coach id 1201.

CREATE OR REPLACE VIEW COACH_VIEW AS SELECT COACH_NAME, TYPE FROM COACH WHERE COACH_ID=1201;

SELECT * FROM COACH_VIEW;

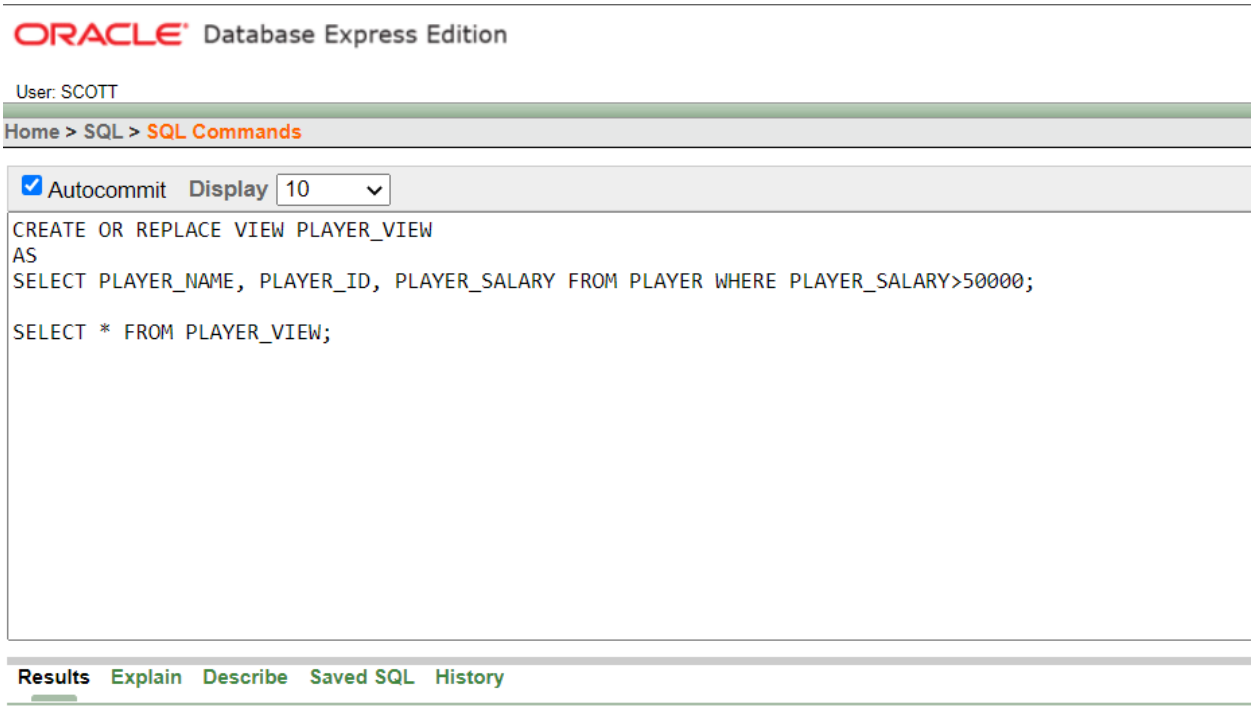




03. Create a view to show the name, id and salary of player who got salary more than 50000

CREATE OR REPLACE VIEW PLAYER_VIEW AS SELECT PLAYER_NAME, PLAYER_ID, PLAYER_SALARY FROM PLAYER WHERE PLAYER_SALARY>50000;

SELECT * FROM PLAYER_VIEW;



Results Explain Describe Saved SQL History

PLAYER_NAME	PLAYER_ID	PLAYER_SALARY
MASHRAFI MORTAZA	1001	80000
LITON DAS	1051	65000
MUSTAFIZUR RAHMAN	1101	70000
TAMIM IQBAL	1151	80000
MAHMUDULLAH	1251	80000

5 rows returned in 0.00 seconds CSV Export

04. Display the CLUB all information along with their respective EMAIL.

```
CREATE OR REPLACE VIEW OWNER_CLUB_VIEW AS
SELECT C.*,O.EMAIL
FROM CLUB C,OWNER O WHERE C.CLUB_ID=O.CLUB_ID;
SELECT * FROM OWNER_CLUB_VIEW;
```

ORACLE Database Express Edition

User: SCOTT

Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE OR REPLACE VIEW OWNER_CLUB_VIEW AS
SELECT C.*,O.EMAIL
FROM CLUB C,OWNER O WHERE C.CLUB_ID=O.CLUB_ID;


SELECT * FROM OWNER_CLUB_VIEW;
```

Results Explain Describe Saved SQL History

View created.

0.49 seconds

Results Explain Describe Saved SQL History

CLUB_ID	CLUB_NAME	DATE_OF_ESTABILISHED	CLUB_PHONE_NUMBER_1	CLUB_PHONE_NUMBER_2	CLUB_PHONE_NUMBER_3	EMAIL
1001	DHAKA PLATOON	2015	94652641	58412658	179654166	bcbimca@gmail.com
1051	RAJSHAHI ROYELS	2012	9545463	1785621265	1941635126	bcb@gmail.com
1201	BARISHAL BULLS	2012	87486542	986646975	1796554486	bcb@gmail.com
1101	RANGPUR RANGERS	2012	54308523	241574521	1815461265	rafsa@gmail.com

4 rows returned in 0.00 seconds CSV Export

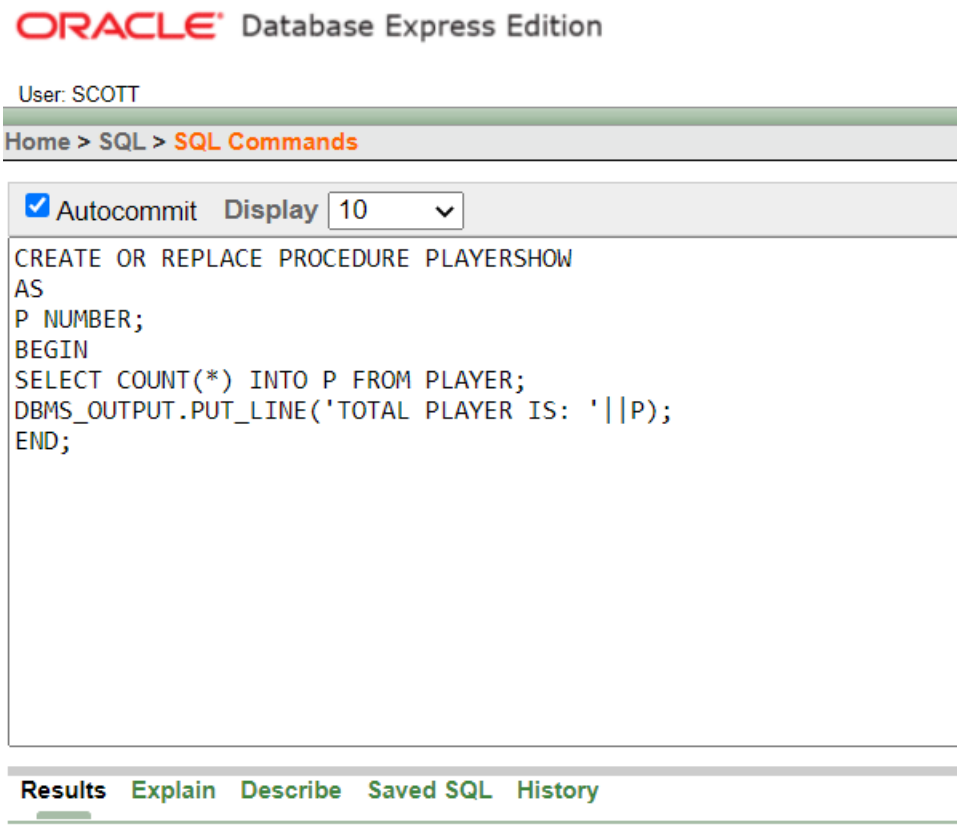
PROCEDURE AND FUNCTION:

PROCEDURE:

01. Write a procedure that will show number of player.

CREATE OR REPLACE PROCEDURE PLAYERSHOW

```
AS
P NUMBER;
BEGIN
SELECT COUNT(*) INTO P FROM PLAYER;
DBMS_OUTPUT.PUT_LINE('TOTAL PLAYER IS: '||P);
END;
```



02. Write a procedure that will show most senior player

```
CREATE OR REPLACE PROCEDURE SENIOR_PLAYER AS
```

P VARCHAR(50);

BEGIN

SELECT PLAYER_NAME INTO P FROM PLAYER WHERE REGESTRATION_DATE=(SELECT MIN(REGESTRATION_DATE)FROM PLAYER);

DBMS_OUTPUT.PUT_LINE('MOST SENIOR PLAYER IS: '||P);

END;

03. Write a procedure that will take as parameter COACH_ID and show coach email.

CREATE OR REPLACE PROCEDURE EMAIL_COACH(CID IN NUMBER) AS

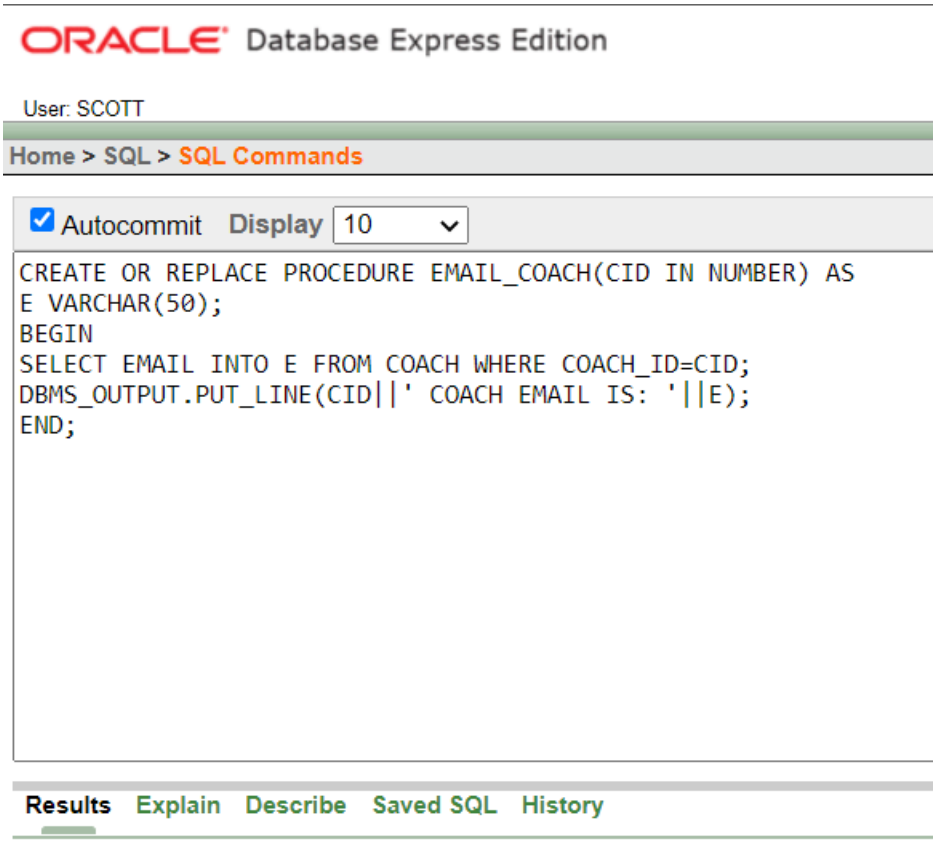
E VARCHAR(50);

BEGIN

SELECT EMAIL INTO E FROM COACH WHERE COACH_ID=CID;

DBMS_OUTPUT.PUT_LINE(CID||' COACH EMAIL IS: '||E);

END;

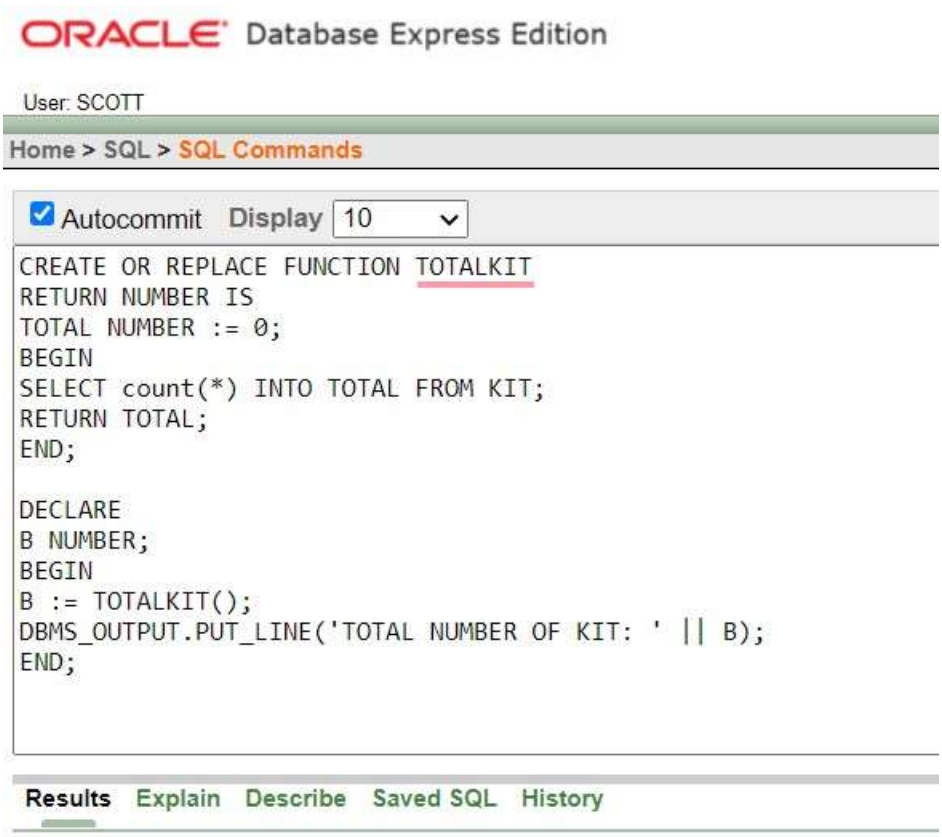


FUNCTION:

04. Create a function that returns the total number of KITS.

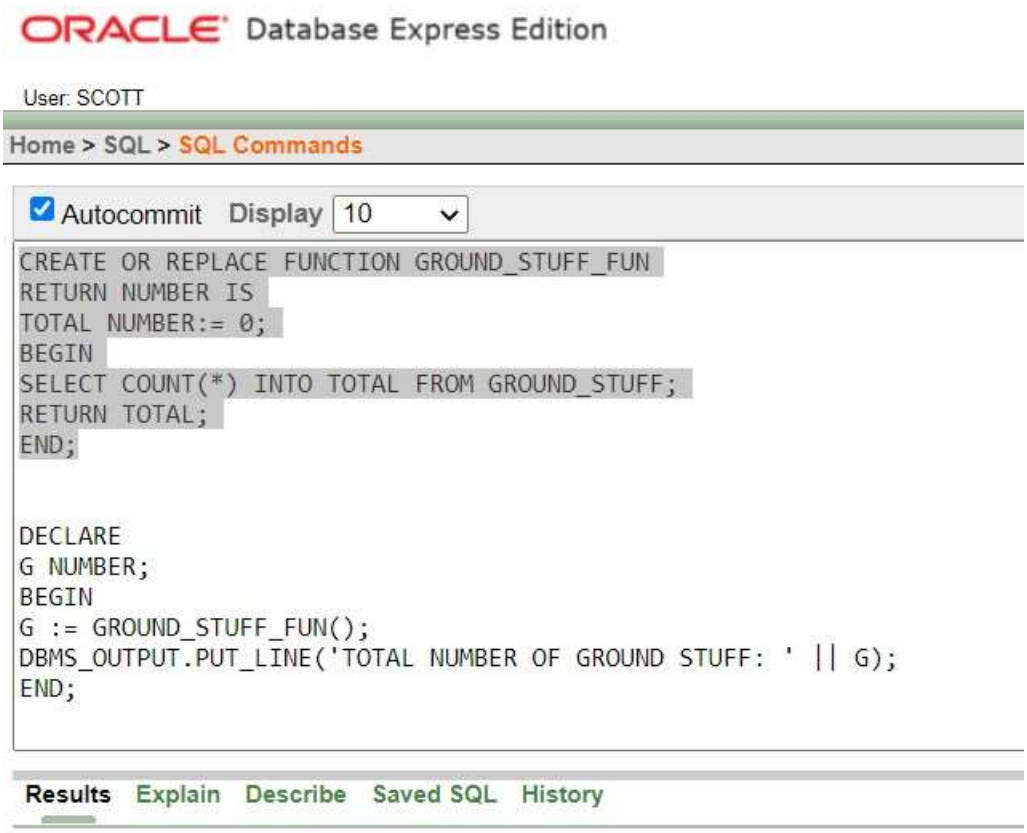
```
CREATE OR REPLACE FUNCTION TOTALKIT

RETURN NUMBER IS

TOTAL NUMBER := 0;

BEGIN

SELECT count(*) INTO TOTAL FROM KIT;

RETURN TOTAL;
```

```
END;


DECLARE
B NUMBER;
BEGIN
B := TOTALKIT();
DBMS_OUTPUT.PUT_LINE('TOTAL NUMBER OF KIT: ' || B);
END;
```



05. Create a function that returns the total number of GROUND STUFF in club.

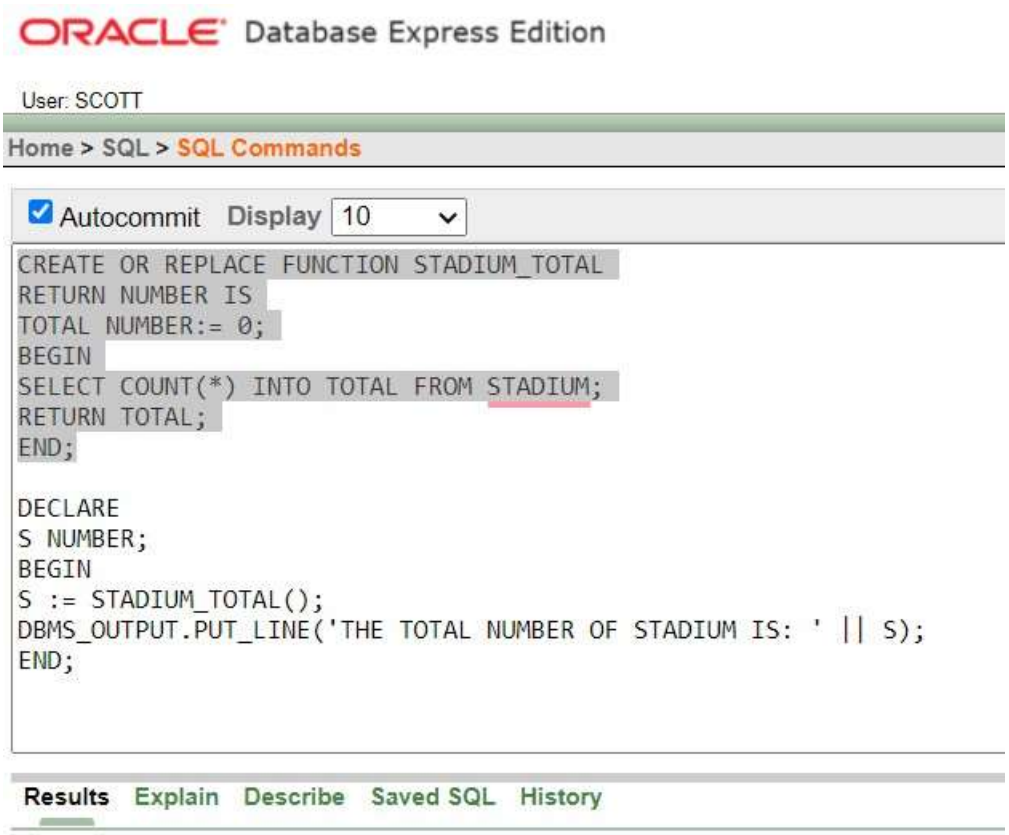
```
CREATE OR REPLACE FUNCTION GROUND_STUFF_FUN
RETURN NUMBER IS
TOTAL NUMBER:= 0;
BEGIN
SELECT COUNT(*) INTO TOTAL FROM GROUND_STUFF;
RETURN TOTAL;
END;


DECLARE
G NUMBER;
BEGIN
G := GROUND_STUFF_FUN();
DBMS_OUTPUT.PUT_LINE('TOTAL NUMBER OF GROUND STUFF: ' || G);
END;
```

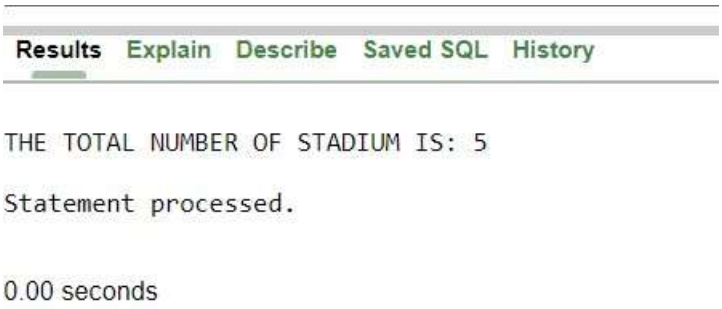


06. Create a function that returns the total number of stadium for club.

```
CREATE OR REPLACE FUNCTION STADIUM_TOTAL
RETURN NUMBER IS
TOTAL NUMBER:= 0;
```

```
BEGIN
SELECT COUNT(*) INTO TOTAL FROM STADIUM;
RETURN TOTAL;
END;

DECLARE
S NUMBER;
BEGIN
S := STADIUM_TOTAL();
DBMS_OUTPUT.PUT_LINE('THE TOTAL NUMBER OF STADIUM IS: ' || S);
END;
```



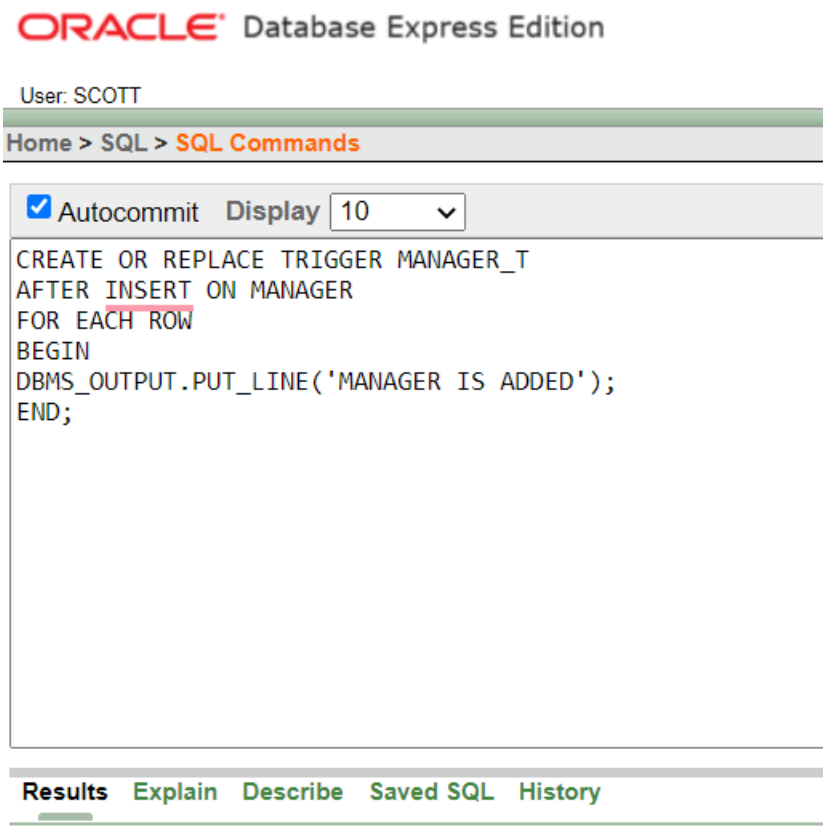


TRIGGER:

01. Create a trigger if insert in manager will print added.

```
CREATE OR REPLACE TRIGGER MANAGER_T

AFTER INSERT ON MANAGER

FOR EACH ROW

BEGIN

DBMS_OUTPUT.PUT_LINE('MANAGER IS ADDED');

END;
```

```
INSERT INTO MANAGER VALUES(1301, 'RAFI', '12-1-2020', 0541855418, 08741895346,
057418954696, 60000);
```

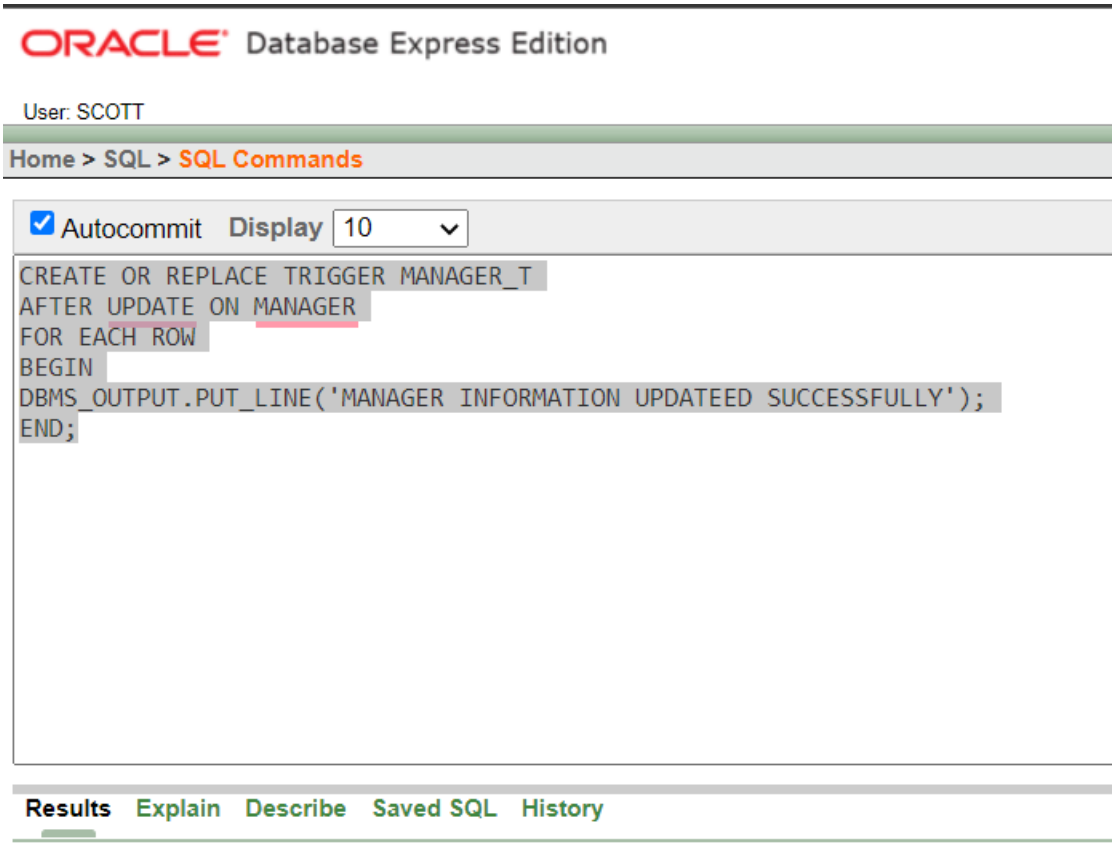


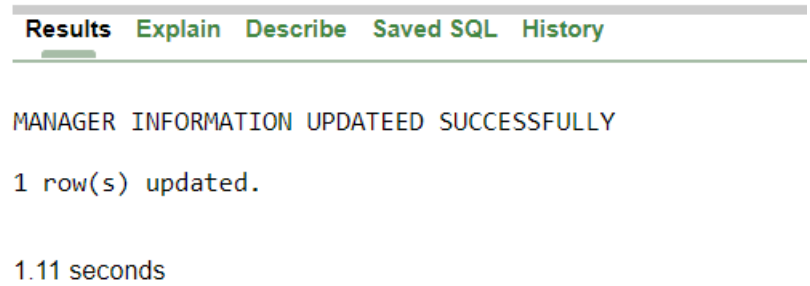

```
Trigger created.

0.03 seconds
```


```
MANAGER IS ADDED

1 row(s) inserted.

0.01 seconds
```

02. Create a trigger if update in manager will print updated.

```
CREATE OR REPLACE TRIGGER MANAGER_T

AFTER UPDATE ON MANAGER

FOR EACH ROW

BEGIN

DBMS_OUTPUT.PUT_LINE('MANAGER INFORMATION UPDATEED SUCCESSFULLY');

END;

UPDATE MANAGER SET MANAGER_SALARY=55000 WHERE MANAGER_ID=1001 ;
```

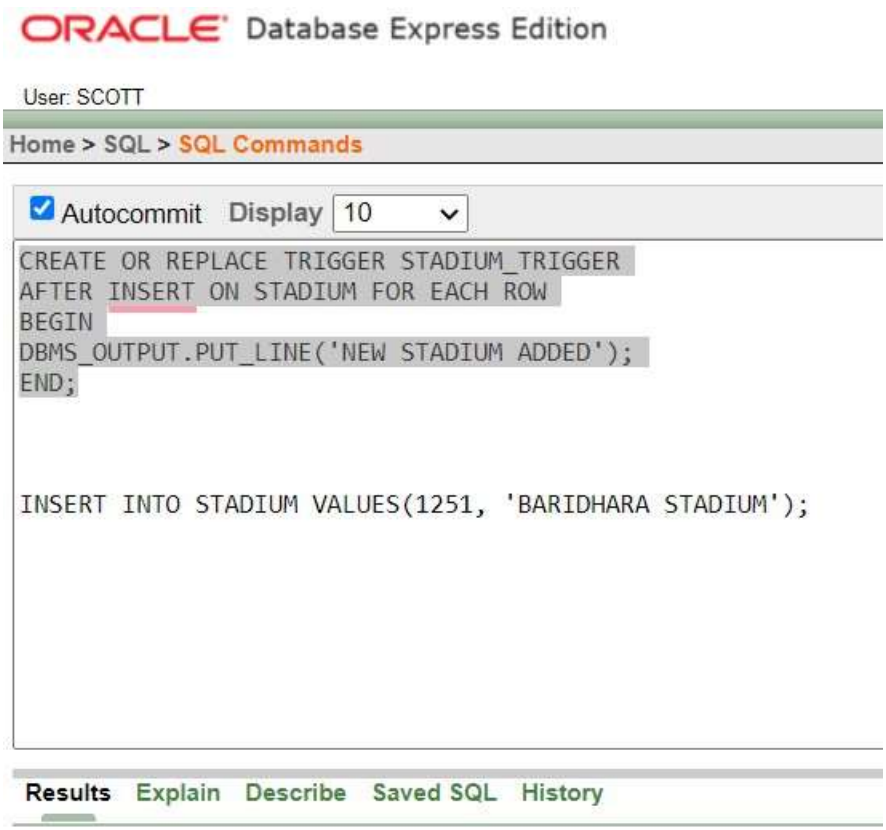




03. Create a trigger if insert in STADIUM will print added.

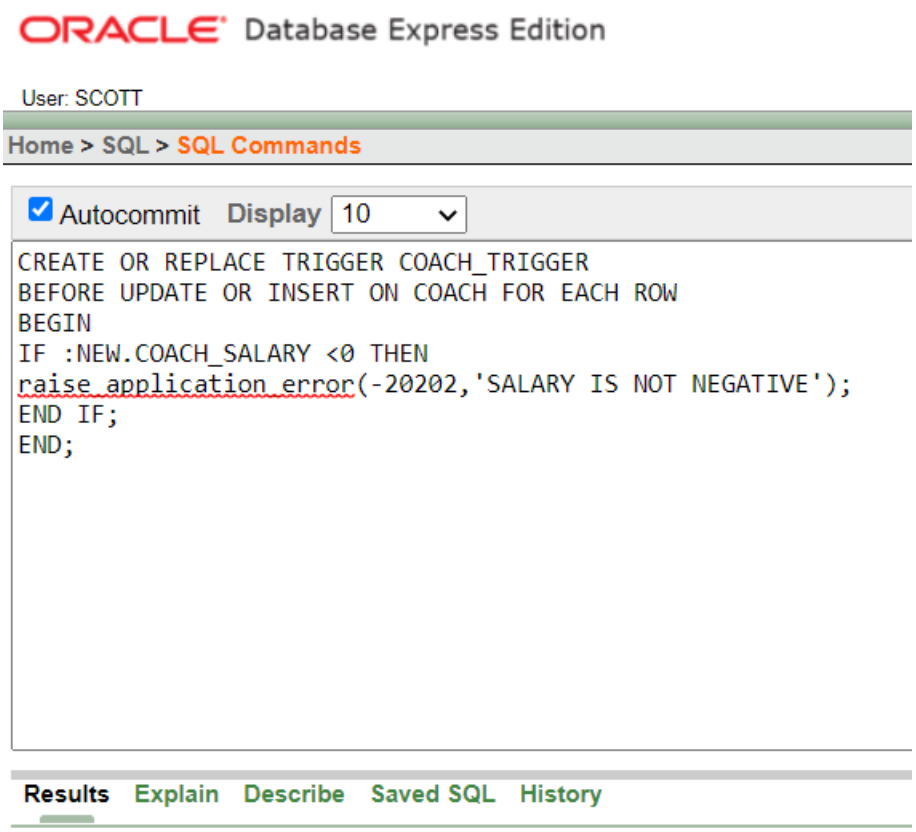
```
CREATE OR REPLACE TRIGGER STADIUM_TRIGGER
AFTER INSERT ON STADIUM FOR EACH ROW
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE('NEW STADIUM ADDED');

END;


INSERT INTO STADIUM VALUES(1251, 'BARIDHARA STADIUM');
```



04. Write triggers that checks that COACH_SALARY does not become negative.

```
CREATE OR REPLACE TRIGGER COACH_TRIGGER
BEFORE UPDATE OR INSERT ON COACH FOR EACH ROW
BEGIN
IF :NEW.COACH_SALARY <0 THEN
raise_application_error(-20202,'SALARY IS NOT NEGATIVE');
END IF;
END;
```




```
Trigger created.

0.00 seconds
```

CONCLUSION :

We have shown all the queries to create the tables in ‘oracle 10g’. We had shown the queries to insert the values and took their screen-shots. The Admin panel is crated and the interface is properly communicated with the oracle database. We have successfully done the procedure and function with the help of plsql. In this system, the trigger is also implemented and the package is applied.

IN THE FUTURE :

This job of ours can help a Cricket club in their data storing systems. Also We Create Users For Easy Our Work. And like the club, we want to make the admin interface for every table so that it can be easy to add, update, and delete data.