**Operations**

**Adding Entry in DAILY\_IN\_OUT\_Out\_Time :**

CALL checkpass\_add\_in\_out\_out(curdate(),1203,'syco’, curtime(),'Malvan','Om@321');

**Adding Entry in DAILY IN\_OUT Add IN Time:**

call checkpass\_add\_in\_out\_In(1203, curdate(),curtime(),'Om@321');

**Register Admin:** INSERT into admin values('Indalkar','I@12');

**STUDENT\_REGISTER\_DATA Add Student Entry:**

CALL Register\_Student(12,'Shreya Sandip Dhamapurkar','OBC','2005-11-24','Kudal', '9423264414', 90000, 'syco', curDATE(), '2358723766','Shreya','Indalkar','I@12');

**Remove Student From STUDENT\_REGISTER\_DATA:**

Call Remove\_Student(1206,'Shreya Sandip Dhamapurkar','Indalkar','I@12');

**See Student In STUDENT\_REGISTER\_DATA TABLE(By Id AND Name):**

Call See\_Student(1205,'SANDesh Dattatray Gade','Indalkar','I@12');

**Searching Student by Initial from STUDENT\_REGISTER DATA:**  Call See\_Student\_By\_initial('o%','Indalkar','I@12');

**Searching Student by Cast from STUDENT\_REGISTER DATA:** Call See\_Student\_By\_caste('OBC','Indalkar','I@12');

**Searching Student by Standard from STUDENT\_REGISTER DATA:**

Call See\_Student\_By\_standard('syco','Indalkar','I@12');

**Searching Student by Birthdate from STUDENT\_REGISTER DATA:**

Call See\_Student\_By\_Birth\_Date('2005-09-01','2006-01-31','Indalkar','I@12');

**Searching Student by Joining\_Date from STUDENT\_REGISTER DATA:**

Call See\_Student\_By\_Join\_Date('2023-09-01','2023-10-22','Indalkar','I@12');

**Searching Student by Income from STUDENT\_REGISTER DATA:**

Call See\_Student\_By\_income(30000,60000,'Indalkar','I@12');

**Searching Student by Address from STUDENT\_REGISTER DATA:** Call See\_Student\_By\_address('Pune%','Indalkar','I@12');

**Updating Student\_edu\_ by id from STUDENT\_REGISTER DATA:** Call UpDATE\_Student\_Edu\_By\_id(1201,'SYCO','Indalkar','I@12');

**Updating Student\_In\_Out\_Passcode\_ by id from STUDENT\_REGISTER DATA:**

Call Update\_Student\_In\_out\_Passcode\_By\_id(1201,'Om@31','Indalkar','I@12');

**Updating Student\_Address\_ by id from STUDENT\_REGISTER DATA:**

Call Update\_Student\_Address\_By\_id(1201,’hadpasar’,'Indalkar','I@12');

**Updating Student\_Family\_income\_ by id from STUDENT\_REGISTER DATA:**

Call Update\_Student\_Family\_income\_By\_id(1201,80000,'Indalkar','I@12');

**Select\_ALL\_ from STUDENT\_REGISTER DATA:**

Call Select\_ALL\_Student\_Regi\_Data('Indalkar','I@12');

**Select\_ALL\_ COUNT from STUDENT\_REGISTER DATA:**

Call Select\_ALL\_Student\_Count\_Regi\_Data('Indalkar','I@12');

* **Creating Database/TABLEs etc. :-**

CREATE DATABASE Hostel\_student\_data; //CREATED Database

USE Hostel\_student\_data;

**STUDENT\_REGISTER\_DATA TABLE:**

CREATE TABLE Student\_Register\_Data(

Sr\_no INT,

Student\_id INT PRIMARY KEY AUTO\_INCREMENT,

Student\_Name VARCHAR(100) NOT NULL,

Caste\_Category VARCHAR(5) check (Caste\_Category='OBC' or Caste\_Category='SC' or Caste\_Category='VJNT' or Caste\_Category='OPEN' or Caste\_Category='NT-D' or Caste\_Category='ST' or Caste\_Category='SBC' or Caste\_Category='NT-B' or Caste\_Category='NT-C') ,

Student\_BirthDATE DATE,

Address VARCHAR(200),

Mobile\_no VARCHAR (10),

Family\_income INT check(Family\_income<100000 AND Family\_income is NOT NULL),

Education VARCHAR(70),

Joining\_DATE DATE,

Account\_no VARCHAR(19),

In\_out\_Passcode VARCHAR(8) );

//CREATED Student\_Register\_Data

ALTER TABLE Student\_Register\_Data AUTO\_INCREMENT =1200;

//SET STARING VALUE FOR INCREMENT

**DAILY\_IN\_OUT TABLE:**

CREATE TABLE Daily\_in\_out (

Sr\_no INT PRIMARY KEY AUTO\_INCREMENT,

Today\_DATE DATE,

Student\_id INT REFERENCES student\_register\_data (Student\_id),

StANDard VARCHAR(70) REFERENCES student\_register\_data (Education),

Time\_of\_Departure TIME,

Reason VARCHAR(100),

Time\_of\_Arrival TIME

);

**For Admin TABLE (Register Pass):**

CREATE TABLE admin(warden VARCHAR(8),pass VARCHAR(4));

insert into admin values('Indalkar','I@12');

* **PROCEDURES:**

1. **Procedure For Daily in Out Add Out time Student Entry:**

DELIMITER //

CREATE PROCEDURE checkpass\_add\_in\_out\_out(

IN Today\_DATE DATE,

IN id INT,

IN Standard VARCHAR (70),

IN Time\_of\_Departure TIME,

IN Reason VARCHAR(100),

IN Pass VARCHAR(8)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM Student\_Register\_Data WHERE Student\_id = id AND In\_out\_Passcode = Pass);

IF pass\_match > 0 THEN

INSERT INTO Daily\_in\_out(Today\_DATE, Student\_id, StANDard, Time\_of\_Departure, Reason)

VALUES (Today\_DATE,id, Standard, Time\_of\_Departure, Reason);

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Register/Password Incorrect: No Student Found';

END IF;

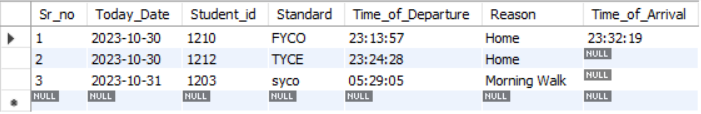
END //

DELIMITER ;

**Adding OUT Entry in** **DAILY\_IN\_OUT TABLE:**

CALL checkpass\_add\_in\_out\_out(curDATE(),1203,'syco',curtime(),'Morning Walk','Om@321'); //No Error

**Result:**

****

**Out Entry Added!**

**Password Correct!**

1. **Procedure For Daily in Out Add In time Student Entry:**

DELIMITER //

CREATE PROCEDURE checkpass\_add\_in\_out\_In(

IN in\_id INT,

In TDATE DATE,

IN in\_Time TIME,

IN in\_Pass VARCHAR(8))

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM Student\_Register\_Data WHERE Student\_id =in\_id AND In\_out\_Passcode = in\_Pass);

IF pass\_match > 0 THEN

UPDATE Daily\_in\_out SET Time\_of\_Arrival = in\_Time WHERE Student\_id=in\_id AND Time\_of\_Arrival is null AND Today\_DATE=TDATE ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Register/Password Incorrect: No Student Found';

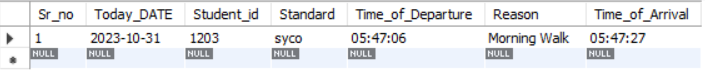
END IF;

END //

DELIMITER ;

**Adding IN Entry in** **DAILY\_IN\_OUT TABLE:**

call checkpass\_add\_in\_out\_In(1203,curDATE(),curtime(),'Om@321');

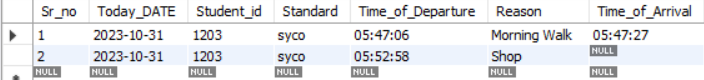
**Result:**

**In Entry Added!**

**Password Correct!**

Wrong Password error!

When call checkpass\_add\_in\_out\_In(1203, curDATE(),curtime(),'Om@32'); //Incorrect Password





1. **Procedure For Adding Student Entry in STUDENT\_REGISTER\_DATA:**

DELIMITER //

CREATE PROCEDURE Register\_Student(

IN Sr\_no INT,

IN Student\_Name VARCHAR(100),

IN Caste\_Category VARCHAR(5) ,

IN Student\_Birthdate DATE,

 IN Address VARCHAR(200),

 IN Mobile\_no VARCHAR (10),

IN Family\_income INT ,

IN Education VARCHAR(70),

IN Joining\_Date DATE,

IN Account\_no VARCHAR(19),

IN In\_out\_Passcode VARCHAR(8),

**Registered!**

**Password Correct!**

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

insert into Student\_Register\_Data

(Sr\_no, Student\_Name, Caste\_Category, Student\_Birthdate, Address,Mobile\_no,Family\_income ,Education,Joining\_Date ,Account\_no, In\_out\_Passcode)

values (Sr\_no,Student\_Name,Caste\_Category,Student\_Birthdate,Address,Mobile\_no,Family\_income,Education,Joining\_Date,Account\_no,In\_out\_Passcode);

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

END //

DELIMITER ;

**Adding Entry in** **STUDENT\_REGISTER\_DATA TABLE:**

CALL Register\_Student(14,'Tanmay Mungekar','SC','2005-11-24','Devgad', '9677866789', 90000, 'TYME', curdate(), '2358723766','Tamu@1','Indalkar','I@12');

1. **Remove Student From**  **STUDENT\_REGISTER\_DATA TABLE:**

DELIMITER //

CREATE procedure Remove\_Student(

IN id INT,

IN Name VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

DELETE from Student\_Register\_Data where Student\_id= id AND Student\_Name=Name;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

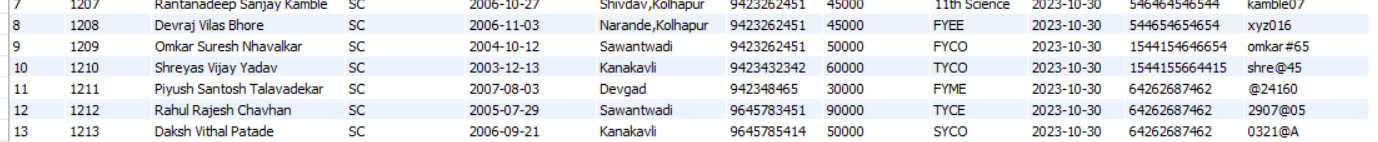
END //

DELIMITER ;

**Removing Student From**  **STUDENT\_REGISTER\_DATA TABLE:**

Call Remove\_Student(1214,'Tanmay Mungekar','Indalkar','I@12');





**Entry Deleted from Student\_Register\_Data Table where Student\_id=1214 and Name = Tanmay Mungekar**

1. **See Student In STUDENT\_REGISTER\_DATA TABLE(By Id AND Name):**

DELIMITER //

CREATE procedure See\_Student(

IN id INT,

IN Name VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Student\_id= id or Student\_Name=Name;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

END //

DELIMITER ;

**Calling:**

Call See\_Student(1205,'Sandesh Dattatray Gade','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table where Student\_id=1205 and Name = Sandesh Dattatray Gade**

1. **Searching Student by Initial from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_initial(

IN ini VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Student\_Name like ini;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

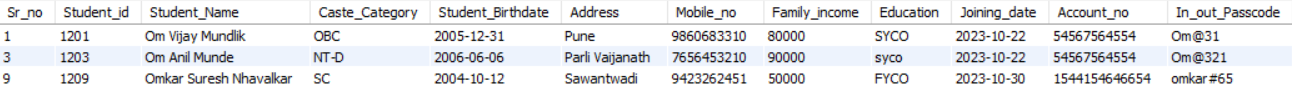
END //

DELIMITER ;

**Calling:**

Call See\_Student\_By\_initial('o%','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table where Name Strat from letter ‘O’**

1. **Searching Student by Cast from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_caste(

IN caste VARCHAR(5),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Caste\_Category=caste ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

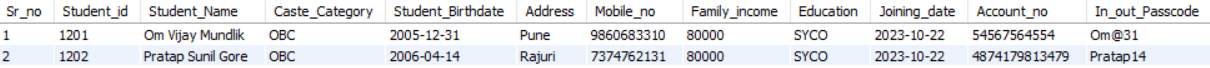
END //

DELIMITER ;

**Calling:**

Call See\_Student\_By\_caste('OBC','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table where Caste Category=OBC**

1. **Searching Student by Standard from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_standard(

IN standard VARCHAR(70),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Education=standard;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

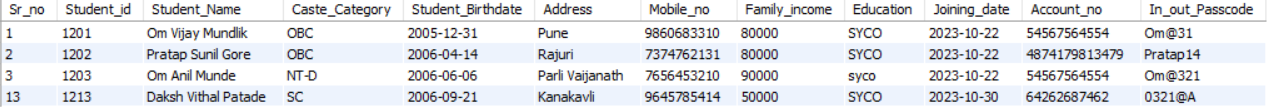
END IF;

END //

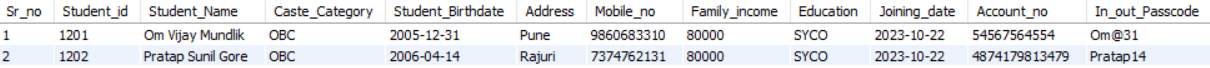
DELIMITER ;

**Calling:**

Call See\_Student\_By\_standard('syco','Indalkar','I@12');



**Searching Student from Student\_Register\_Data Table where Education=SYCO**



1. **Searching Student by Birthdate from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_Birth\_Date(

IN here DATE,

IN there DATE,

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Student\_Birthdate between here AND there ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

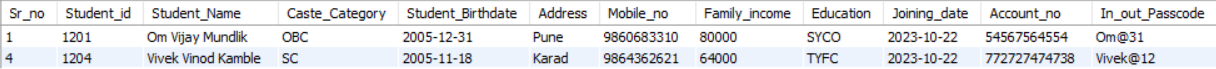
END //

DELIMITER;

**Calling:**

Call See\_Student\_By\_Birth\_DATE('2005-09-01','2006-01-31','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table whose Birthdate between '2005-09-01' to '2006-01-31'**

1. **Searching Student by Joining Date from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_Join\_Date(

IN here DATE,

IN there DATE,

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Joining\_Date between here AND there ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

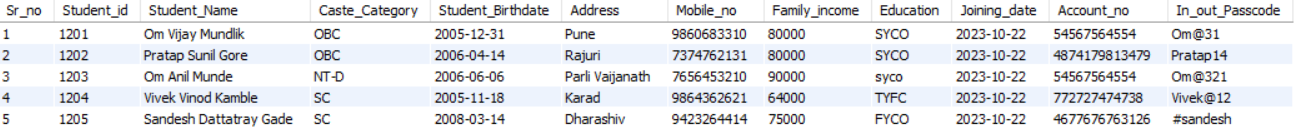
END //

DELIMITER ;

**Calling:**

Call See\_Student\_By\_Join\_Date('2023-09-01','2023-10-22','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table who Join between '2023-09-01'to '2023-10-22'**

1. **Searching Student by Income from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_income(

IN here INT,

IN there INT,

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Family\_income between here AND there ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

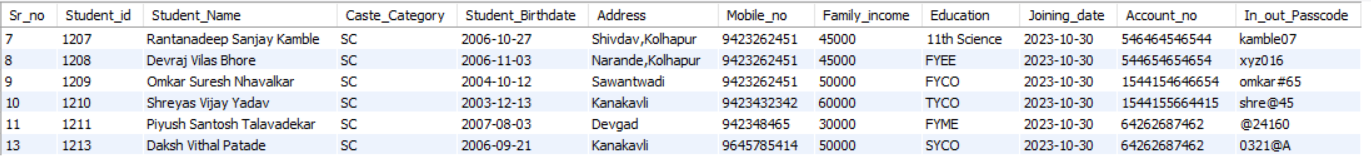
END //

DELIMITER ;

**Calling:**

Call See\_Student\_By\_income(30000,60000,'Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table whose Family Income between 30000 to 60000**

1. **Searching Student by Address from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure See\_Student\_By\_address(

IN home VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data where Address like home ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

END //

DELIMITER ;

**Calling:**

Call See\_Student\_By\_address('Pune%','Indalkar','I@12');





**Searching Student from Student\_Register\_Data Table where Address contains Pune**

1. **Updating Student\_edu\_ by id from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure update\_Student\_Edu\_By\_id(

IN id INT,

IN edu VARCHAR(70),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

UPDATE student\_register\_data set Education=edu where Student\_id=id ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

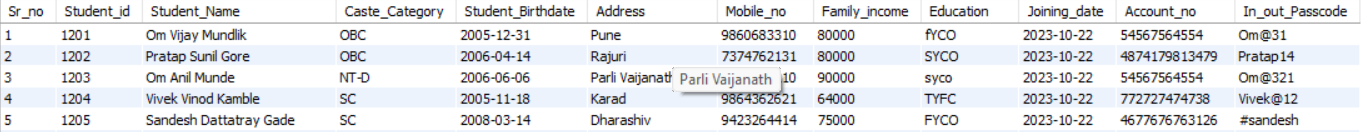
END //

DELIMITER ;

**Calling:**

Call Update\_Student\_Edu\_By\_id(1201,'fYCO','Indalkar','I@12');





**Updating Student from Student\_Register\_Data Table as Education=fYCO where Student\_id=1201**

1. **Updating Student\_In\_Out\_Passcode\_ by id from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure update\_Student\_In\_Out\_Passcode\_By\_id(

IN id INT,

IN passw VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

UPDATE student\_register\_data set In\_out\_Passcode=passw where Student\_id=id ;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

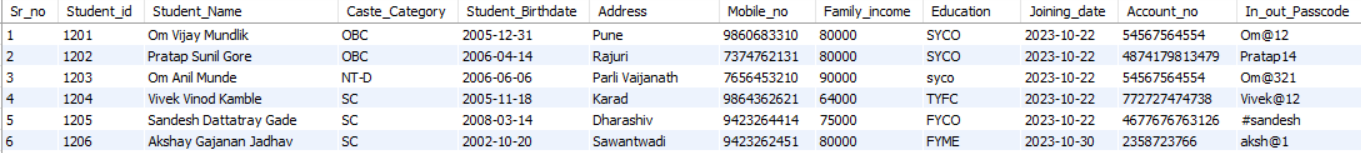
END //

DELIMITER ;

**Calling:**

Call Update\_Student\_In\_out\_Passcode\_By\_id(1201,'Om@31','Indalkar','I@12');





**Updating Student from Student\_Register\_Data Table as In\_out\_Password = Om@12 where Student\_id=1201**

1. **Updating Student\_Address\_ by id from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure update\_Student\_Address\_By\_id(

IN id INT,

IN home VARCHAR(100),

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

UPDATE student\_register\_data set Address=home where Student\_id=id;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

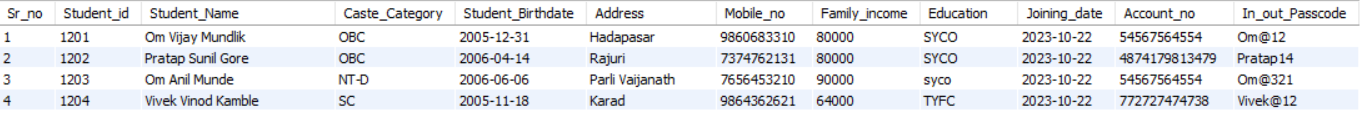
END IF;

END //

DELIMITER ;

**Calling:**

Call Update\_Student\_Address\_By\_id(1201,’hadpasar’,'Indalkar','I@12');



**Updating Student from Student\_Register\_Data Table as Address= Hadapsar where Student\_id=1201**

1. **Updating Student\_Family\_income\_ by id from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure update\_Student\_Family\_Income\_By\_id(

IN id INT,

IN income INT,

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

UPDATE student\_register\_data set Family\_income=income where Student\_id=id;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

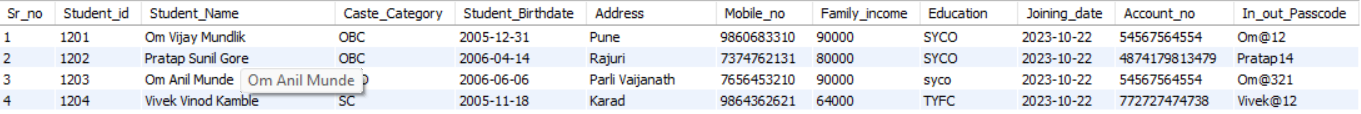
END IF;

END //

DELIMITER ;

**Calling:**

Call Update\_Student\_Family\_income\_By\_id(1201,80000,'Indalkar','I@12');



**Updating Student from Student\_Register\_Data Table as Family income = 90000 where Student\_id=1201**

1. **Select\_ALL\_ from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure Select\_ALL\_Student\_Regi\_Data(

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select \* from Student\_Register\_Data;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

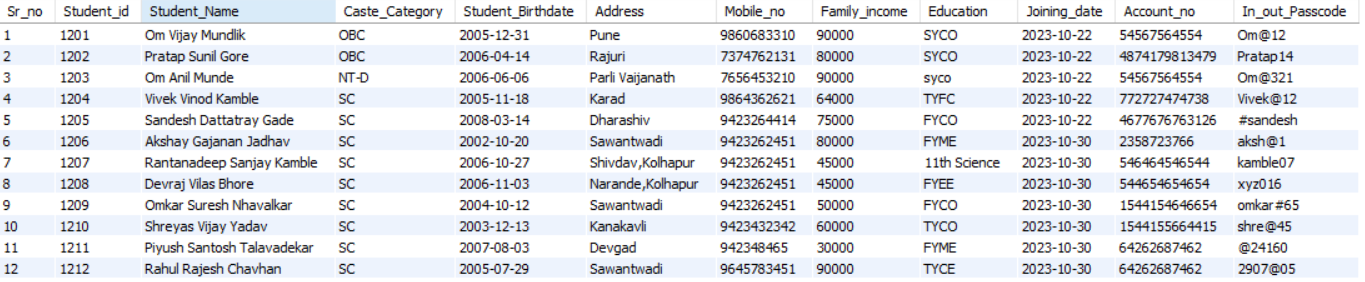
END //

DELIMITER ;

**Calling:**

call Select\_ALL\_Student\_Regi\_Data('Indalkar','I@12');





**Display All Register Students**

1. **Select\_ALL\_ COUNT from STUDENT\_REGISTER DATA:**

DELIMITER //

CREATE procedure Select\_ALL\_Student\_Count\_Regi\_Data(

IN Admin VARCHAR(8),

IN Admin\_Pass VARCHAR(4)

)

BEGIN

DECLARE pass\_match INT;

SET pass\_match = (SELECT COUNT(\*) FROM admin WHERE warden= Admin AND pass = Admin\_Pass);

IF pass\_match > 0 THEN

Select count(\*) as Total\_Student from Student\_Register\_Data;

ELSE

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Denied Access: User/Password incorrect';

END IF;

END //

DELIMITER ;

Calling:

call Select\_ALL\_Student\_Count\_Regi\_Data('Indalkar','I@12');





**Display Total Register Students**

* **ER DIAGRAM:**

Mobile\_No

Address

Reason

Time\_of\_Arrival

Time\_of\_Departure

Standard

Student\_id

Today\_Date

Sr. No

Refers

Pass

Warden

Handled By

Sr. No

In\_out\_Passcode

Account\_No

Joining\_date

Education

Family\_income

Student\_Birthdatee

Caste\_Category

Student\_Name

Student\_id

Student\_Register\_Data

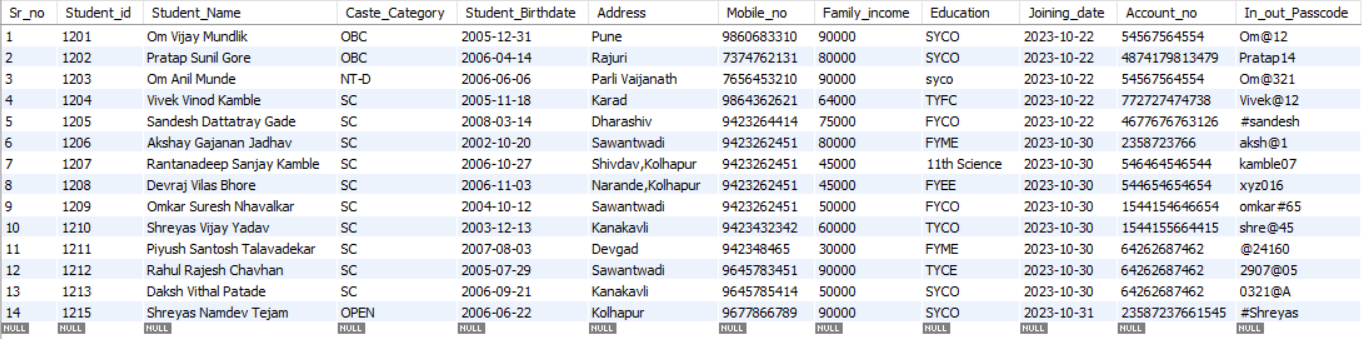
Sr. No

Daily\_in\_out

Admin

* **TABLES:**

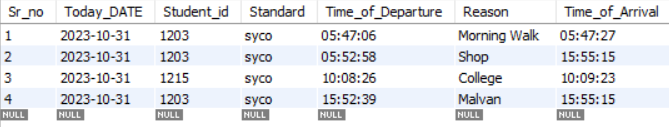
1. **Student\_Register\_Data**

****

1. **Admin**

****

1. **Daily\_in\_out**

****

* **FEATURES:**

1. **Database Creation:**

* A database named "Hostel\_student\_data" is created.
* Tables include "Student\_Register\_Data," "Admin," and "Daily\_in\_out."
* Appropriate constraints and relationships are defined in the tables.

1. **Stored Procedures:**

* Several stored procedures are created to interact with the database and manage student records, admin access, and daily in-out entries.

1. **Admin Authentication:**

* Admin authentication is required for various operations.
* The admin's "warden" and "pass" are checked to ensure only authorized users can perform operations.

1. **Student Registration:**

* Students can be registered with details such as name, caste category, birthdate, address, mobile number, family income, education, joining date, account number, and in-out passcode.
* Admin authentication is required for student registration.

1. **Student Removal:**

* Students can be removed from the database by providing the student's ID and name.
* Admin authentication is required for student removal.

1. **Student Information Retrieval:**

* Students' information can be retrieved by their ID or name.
* Admin authentication is required for information retrieval.

1. **Search by Initial, Caste, Standard, Birthdate, Joining Date, Income, and Address:**

* Students can be searched based on various criteria, including initial characters of their name, caste, standard, birthdate range, joining date range, income range, and address.
* Admin authentication is required for these searches.

1. **Updating Student Information:**

* Student information such as education, in-out passcode, address, and family income can be updated by providing the student's ID.
* Admin authentication is required for updates.

1. **Total Student Count:**

* The total count of students in the database can be retrieved.
* Admin authentication is required for this operation.

1. **Daily In-Out Entries:**

* Daily in-out entries can be recorded for students with details such as date, student ID, standard, departure time, reason, and arrival time.
* Admin authentication is required for recording in-out entries.

1. **Error Handling:**

* The code includes error handling to check for correct admin credentials and student passcodes before performing actions.

1. **ER Diagram:**

* The program includes an ER (Entity-Relationship) diagram to visualize the database schema and relationships between tables.