GROUP PROJECT - 4

INTRODUCTION:

Imagine managing visa applications with scattered spreadsheets and disconnected systems. This is the struggle that US immigration and various related organizations face today, leading to **Inefficiency** as manually processing visa applications for dependents and tourists, tracking financial aid, and managing course enrollment for international students is time-consuming and error prone.

Our database application describes the immigration process and specifically three types of visas. The USCIS (United States Citizenship and Immigration Services) controlling visa approval is essential to this system because they not only control immigration and stay but also have a significant influence on immigrants' academic and career paths. The focus is placed on international students as has a very significant presence in the database, so the database is further extended to maintain records of their employment in the university, and courses taken. It is also essential to note that the two other visas mentioned in the ER diagram below are also of significance and at a later stage effort will be made to reduce data redundancy and make the database as operationally efficient.

We aim to create a centralized system for managing visa applicants' data, driven by the following motivations:

- **Streamline Processes**: **Automate** tasks like visa applications, financial aid disbursement, and course registration, freeing up staff time and resources.
- **Improve Data Accuracy**: **Eliminate** manual data entry and ensure **consistent** information across the institution.

The potential benefits are:

- **Increased Operational Efficiency**: **Streamlined** workflows and centralized data lead to faster processing times and reduced administrative burden.
- **Data-Driven Strategies**: Insights gained from the database empower informed decision-making regarding international student recruitment, program development, and support services for visa applicants.

This database will be a valuable tool for various users, including:

- **International Student Advisors**
- **Financial Aid Officers**
- **University Administration**
- **Homeland Security**
- **USCIS (U.S Citizenship and immigration services)**
- **SEVIS (Student and Exchange Visitor Information System)**

By centralizing and organizing international student data, this database has the potential to transform how institutions manage their international student population, leading to a more efficient, informed, and studentcentric experience as well as managing the visa applications of dependent and tourist visa applicants.

BUSSINESS RULES:

1. Financial Aid Management:

Financial Aid Records: A FinancialAidID individually identifies each financial aid record (Scholarship, Loan, Sponsor) and associates it with a particular student.

Types of Financial Aid: A variety of financial assistance options are available to students, such as CGPA-based scholarships, loans from certain lenders with grace periods, and sponsorships that indicate the sponsor's complete identity and affiliation to the student.

2. Student and Visa Management:

Visa and Student Linkage: Since the Passport_ID is utilized in both the Student and Visa tables, each student must be connected with a valid visa**.**

Visa Validity and Dependents: Important information such as visa, its expiration date, and the applicant's personal data must be included in the visa records. A relationship between the dependents and the visa holder must be managed within the system as well.

3. Course Enrollment and Management:

Course Sections: Every course consists of one or more sections, each distinguished specifically within a given year and semester.

Student Enrollment: Students are able to enroll in more than one part, and an Enrollment ID is used to monitor each student's enrollment information separately.

4. Employment Tracking: Employment within the university (by department and position), must be related to students and have beginning and ending dates.

5. Tourist Visa Handling: People holding a tourist visa are monitored individually, including fields for the length of stay and the anticipated arrival date.

USER REQUIREMENTS:

1. Comprehensive Data Access:

Students: Must have access to their work records, financial aid status, visa information, course enrollments, and personal data.

Administrative Staff: For compliance and operational reasons, require access to all student data, financial assistance disbursements, course and section administration interfaces, and reporting tools.

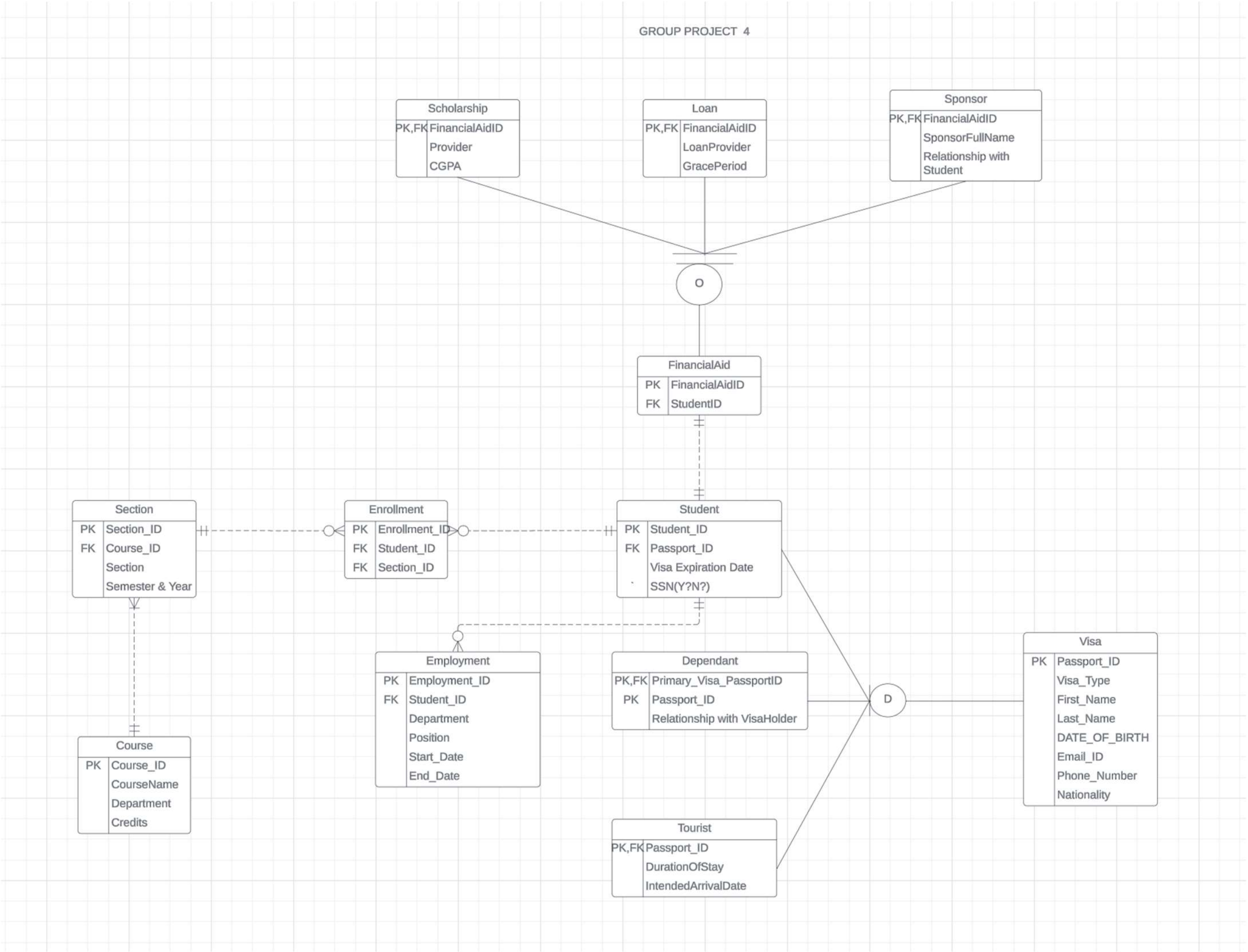
2. Security and Data Privacy: The system needs to protect sensitive and private information while abiding by privacy laws and regulations. To protect the visibility and manipulability of data based on user roles, role-based access controls are part of this.

3. Reporting and Analytics: To support regulatory compliance and decision-making, the system should have strong reporting capabilities for producing reports on employment data, financial assistance, visa statuses, and enrollment.

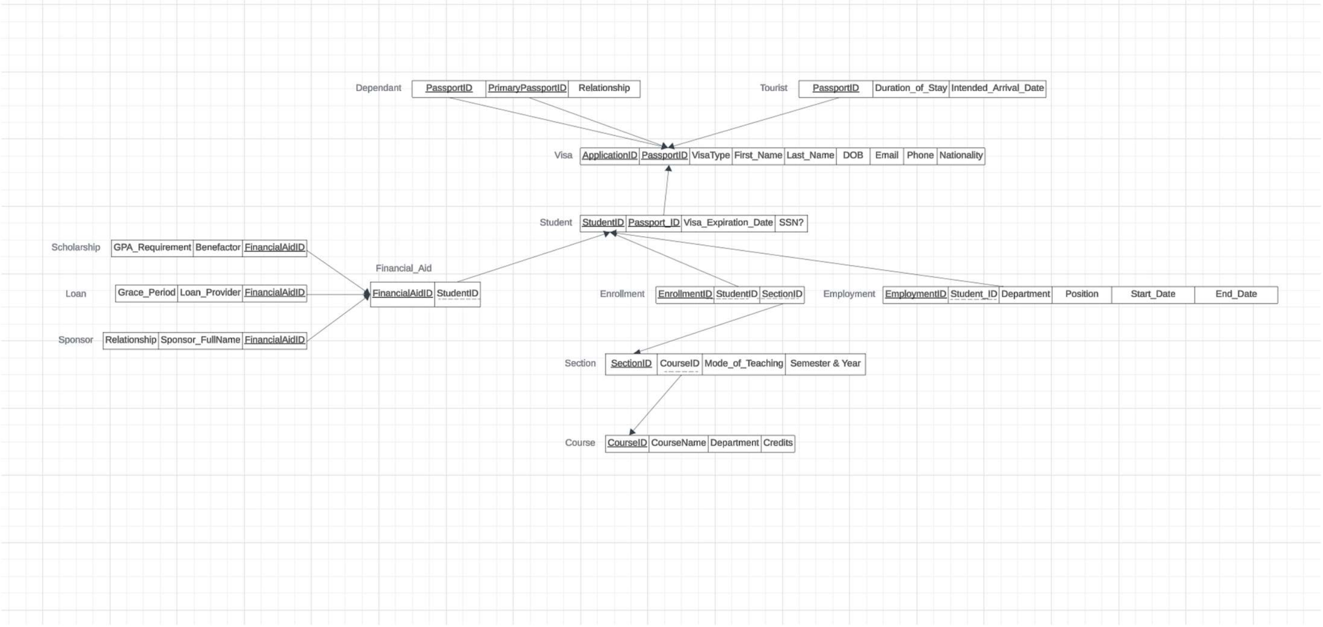
4. Real-Time Notifications and Alerts: Notifications about enrollment deadlines, financial assistance application periods, and visa expiration dates should be sent to users by the system. Administrative employees must also get alerts about papers that are about to expire or compliance problems.

5. Maintenance and Support: In order to fix issues, update features, and adjust to new specifications or modifications in compliance standards, systems need constant support and maintenance.

Enhanced Entity Relation Diagram (EERD):



RELATIONAL MODEL:



SCRIPT 1: TABLE CREATION AND INSERTION

```
create table Visa(
    Passport_ID varchar2(255) NOT NULL PRIMARY KEY,
    Visa_Type varchar2(5) NOT NULL,
    First_Name varchar2(50) NOT NULL,
    Last_Name varchar2(50) NOT NULL,
    Date_of_Birth date,
    Email_ID varchar2(255) NOT NULL,
    Phone_Number int NOT NULL,
    Nationality varchar2(50) NOT NULL
);

create table Tourist(
    Passport_ID varchar2(255) NOT NULL PRIMARY KEY,
    Duration_Of_Stay INT,
    Intended_Arrival_Date DATE,
    FOREIGN KEY (Passport_ID) references Visa(Passport_ID)
);

create table Dependant(
    Primary_Passport_ID varchar2(255) NOT NULL,
    Passport_ID varchar(255) NOT NULL,
    Relationship varchar2(10),
    PRIMARY KEY (Passport_ID,Primary_Passport_ID),
    FOREIGN KEY (Primary_Passport_ID) references Visa(Passport_ID)
);

create table Student(
    Student_ID INT NOT NULL,
    Passport_ID varchar2(255) NOT NULL,
    Visa_Expiration_Date Date,
    IS_HAVING_SSN CHAR(1) CHECK (IS_HAVING_SSN IN ('Y','N')),
    PRIMARY KEY (Student_ID),
    FOREIGN KEY(Passport_ID) references Visa(Passport_ID)
);

create table enrollment (
    Enrollment_ID NUMBER NOT NULL PRIMARY KEY,
    Student_ID INT NOT NULL,
    Section_ID  VARCHAR2(255) NOT NULL,
    FOREIGN KEY (Student_ID) references Student(Student_ID)
);

create table section (
    Section_ID varchar2(255) NOT NULL PRIMARY KEY,
    Course_ID INT NOT NULL,
    Section varchar2(255),
    Semester_Year varchar2(255)
);

alter table enrollment
add constraint enroll_fk2 FOREIGN KEY (Section_ID) references section(Section_ID);

create table course (
    Course_ID INT NOT NULL PRIMARY KEY,
    Course_Name char(20),
    Department char(20),
    Credits INT
);

alter table section
```

```
add constraint section_fk1 FOREIGN KEY (Course_ID) references course(Course_ID);

create table FinancialAid (
   FinancialAidID varchar2(255) NOT NULL PRIMARY KEY,
   Student_ID INT NOT NULL,
   FOREIGN KEY (Student_ID) references Student(Student_ID)
);

create table Scholarship (
   FinancialAidID varchar2(255) NOT NULL PRIMARY KEY,
   Provider char(20),
   CGPA INT NOT NULL,
   FOREIGN KEY (FinancialAidID) references FinancialAid(FinancialAidID)
);

create table Loan (
   FinancialAidID varchar2(255) NOT NULL PRIMARY KEY,
   Loan_Provider char(20),
   Grace_Period varchar2(255),
   FOREIGN KEY (FinancialAidID) references FinancialAid(FinancialAidID)
);

create table Sponsor (
   FinancialAidID varchar2(255) NOT NULL PRIMARY KEY,
   Sponsor_FullName char(50),
   Relationship char(20),
   FOREIGN KEY (FinancialAidID) references FinancialAid(FinancialAidID)
);

create table Employment (
   Employment_ID varchar2(255) PRIMARY KEY,
    Student_ID INT,
    Department char(50),
    Position_ varchar2(255),
    Start_Date Date,
    End_Date Date,
    FOREIGN KEY (Student_ID) references Student(Student_ID)
);

INSERT INTO VISA VALUES ('Z3551512','F-1','Rohit','Sharma',to_date('04-30-1997','mm-dd-yyyy'),'rsharmak@gmail.com',5298878666,'India');
INSERT INTO VISA VALUES ('A7045082','F-1','Virat','Kohli',to_date('05-23-1998','mm-dd-yyyy'),'vkohli34@gmail.com',3189685376,'India');
INSERT INTO VISA VALUES ('B1820313','F-1','Yashaswi','Jaiswal',to_date('11-05-1996','mm-dd-yyyy'),'jaiswaly2@gmail.com',2029212914,'India');
INSERT INTO VISA VALUES ('C7003865','B-2','Jasprit','Bumrah',to_date('09-14-2003','mm-dd-yyyy'),'jbumrah96@gmail.com',8228254437,'SouthKorea');
INSERT INTO VISA VALUES ('U3551512','F-1','Kuldeep','Yadav',to_date('06-09-2001','mm-dd-yyyy'),'kyadav45@gmail.com',7802263727,'Japan');
INSERT INTO VISA VALUES ('P7599464','IR2','Shivam','Dube',to_date('04-19-2000','mm-dd-yyyy'),'dshivam34@gmail.com',8783596247,'India');
INSERT INTO VISA VALUES ('J7599479','F-1','Lin','Chen',to_date('11-03-2001','mm-dd-yyyy'),'lin.chen@student.com',8906783451,'China');
INSERT INTO VISA VALUES ('P7599480','F-1','Aisha','Mohamed',to_date('07-14-1997','mm-dd-yyyy'),'aisham@college.net',5673452109,'Egypt');
INSERT INTO VISA VALUES ('L7599481','F-1','Thomas','Schmidt',to_date('03-08-2003','mm-dd-yyyy'),'tschmidt@mail.edu',1230987654,'Germany');
INSERT INTO VISA VALUES ('S7599482','IR2','Christopher','Lee',to_date('09-12-1995','mm-dd-yyyy'),'chris.lee@email.com',4561237890,'South Korea');
INSERT INTO VISA VALUES ('Y7599483','IR2','Anya','Petrova',to_date('02-20-1998','mm-dd-yyyy'),'apetrova@home.com',7890123456,'Bulgaria');
INSERT INTO VISA VALUES ('V7599484','B-2','Isabelle','Blanc',to_date('06-05-1980','mm-dd-yyyy'),'isabelle.blanc@travel.com',3452106789,'France');
INSERT INTO VISA VALUES ('X7599485','B-2','Omar','Hassan',to_date('10-27-1972','mm-dd-yyyy'),'ohassan@tourism.com',9874561230,'Egypt');
INSERT INTO TOURIST VALUES ('C7003865',30,to_date('10-27-2023','mm-dd-yyyy'));
INSERT INTO TOURIST VALUES ('V7599484',30,to_date('10-27-2023','mm-dd-yyyy'));
INSERT INTO TOURIST VALUES ('X7599485', 25, to_date('02-14-2024','mm-dd-yyyy'));

INSERT INTO DEPENDANT VALUES ('Z3551512','P7599464','WIFE');
INSERT INTO DEPENDANT VALUES ('B1820313','S7599482','HUSBAND');
INSERT INTO DEPENDANT VALUES ('L7599481','Y7599483','SON');

INSERT INTO STUDENT VALUES (22931943,'Z3551512',to_date('09-24-2025','mm-dd-yyyy'),'Y');
INSERT INTO STUDENT VALUES (44206825,'A7045082',to_date('04-2-2025','mm-dd-yyyy'),'Y');
INSERT INTO STUDENT VALUES (82601364,'B1820313',to_date('02-16-2026','mm-dd-yyyy'),'N');
INSERT INTO STUDENT VALUES (68391501,'U3551512',to_date('07-13-2025','mm-dd-yyyy'),'N');
INSERT INTO STUDENT VALUES (46840060,'J7599479',to_date('11-5-2026','mm-dd-yyyy'),'Y');
INSERT INTO STUDENT VALUES (82124271,'P7599480',to_date('06-28-2025','mm-dd-yyyy'),'Y');
INSERT INTO STUDENT VALUES (14133421,'L7599481',to_date('01-9-2025','mm-dd-yyyy'),'N');

INSERT INTO COURSE VALUES (5869,'FDM','IST',3);
INSERT INTO COURSE VALUES (5394,'BADS','IST',3);
INSERT INTO COURSE VALUES (4591,'JAVA','COM',3);
INSERT INTO COURSE VALUES (4672,'DSA','COM',3);
INSERT INTO COURSE VALUES (5591,'INFOVIZ','IST',4);
INSERT INTO COURSE VALUES (4436,'COMARCH','COM',4);

INSERT INTO SECTION VALUES ('IST110',5869,'FULTON','SP24');
INSERT INTO SECTION VALUES ('CS110',5394,'CENTENNIAL','FS24');
INSERT INTO SECTION VALUES ('IST125',4591,'FULTON','FS24');
INSERT INTO SECTION VALUES ('CS135',4672,'CENTENNIAL','SP24');
INSERT INTO SECTION VALUES ('IST221',5591,'FULTON','SP24');
INSERT INTO SECTION VALUES ('CS310',4436,'CENTENNIAL','FS24');
INSERT INTO ENROLLMENT VALUES (298026,22931943,'IST110');

INSERT INTO ENROLLMENT VALUES (338356,22931943,'CS110');

INSERT INTO ENROLLMENT VALUES (138497,22931943,'CS135');

INSERT INTO ENROLLMENT VALUES (203477,44206825,'IST125');

INSERT INTO ENROLLMENT VALUES (376445,44206825,'IST110');

INSERT INTO ENROLLMENT VALUES (414644,82601364,'CS135');

INSERT INTO ENROLLMENT VALUES (479363,82601364,'CS110');

INSERT INTO ENROLLMENT VALUES (969272,82601364,'IST110');

INSERT INTO ENROLLMENT VALUES (508239,68391501,'CS310');

INSERT INTO ENROLLMENT VALUES (237427,68391501,'IST125');

INSERT INTO ENROLLMENT VALUES (387947,46840060,'IST221');

INSERT INTO ENROLLMENT VALUES (548578,46840060,'CS110');

INSERT INTO ENROLLMENT VALUES (912866,46840060,'IST110');

INSERT INTO ENROLLMENT VALUES (556227,82124271,'CS135');

INSERT INTO ENROLLMENT VALUES (284277,82124271,'IST221');

INSERT INTO ENROLLMENT VALUES (121033,14133421,'IST125');

INSERT INTO ENROLLMENT VALUES (300189,14133421,'CS310');

INSERT INTO ENROLLMENT VALUES (601366,14133421,'IST221');


INSERT INTO FINANCIALAID VALUES ('RsharmaSecure',22931943);

INSERT INTO FINANCIALAID VALUES ('ViratKohliSecure',44206825);

INSERT INTO FINANCIALAID VALUES ('YJaiswalSecure',82601364);

INSERT INTO FINANCIALAID VALUES ('KYadavSecure',68391501);

INSERT INTO FINANCIALAID VALUES ('LChenSecure',46840060);

INSERT INTO FINANCIALAID VALUES ('AMohamedSecure',82124271);

INSERT INTO FINANCIALAID VALUES ('TSchmidtSecure',14133421);


INSERT INTO EMPLOYMENT VALUES (629218,22931943,'Dining','Barista',to_date('01-03-2023','mm-dd-yyyy'),to_date('04-24-2023','mm-dd-yyyy'));

INSERT INTO EMPLOYMENT VALUES (713385,22931943,'IST','Grader',to_date('02-15-2023','mm-dd-yyyy'),to_date('05-05-2023','mm-dd-yyyy'));
INSERT INTO EMPLOYMENT VALUES (382603,44206825,'Dining','Barista',to_date('03-29-2023','mm-dd-yyyy'),to_date('06-20-2023','mm-dd-yyyy'));

INSERT INTO EMPLOYMENT VALUES (843376,46840060,'Dining','Barista',to_date('04-07-2023','mm-dd-yyyy'),to_date('07-22-2023','mm-dd-yyyy'));

INSERT INTO EMPLOYMENT VALUES (437587,46840060,'COM','Grader',to_date('01-19-2023','mm-dd-yyyy'),to_date('05-26-2023','mm-dd-yyyy'));

INSERT INTO EMPLOYMENT VALUES (549355,82124271,'Dining','Barista',to_date('02-09-2023','mm-dd-yyyy'),to_date('06-02-2023','mm-dd-yyyy'));

INSERT INTO EMPLOYMENT VALUES (259522,82124271,'SSC','FrontDesk',to_date('03-11-2023','mm-dd-yyyy'),to_date('05-18-2023','mm-dd-yyyy'));


INSERT INTO SCHOLARSHIP VALUES ('RsharmaSecure','PROVOST',3.5);

INSERT INTO SCHOLARSHIP VALUES ('ViratKohliSecure','DEAN',3);

INSERT INTO SCHOLARSHIP VALUES ('YJaiswalSecure','PROVOST',3.5);

INSERT INTO SCHOLARSHIP VALUES ('KYadavSecure','DEAN',3);

INSERT INTO SCHOLARSHIP VALUES ('LChenSecure','PROVOST',3.5);

INSERT INTO SCHOLARSHIP VALUES ('AMohamedSecure','DEAN',3);


INSERT INTO SPONSOR VALUES ('ViratKohliSecure','KARN SHARMA','FATHER');

INSERT INTO SPONSOR VALUES ('YJaiswalSecure','ADITHYA JAISWAL','BROTHER');

INSERT INTO SPONSOR VALUES ('KYadavSecure','LAL YADAV','FATHER');

INSERT INTO SPONSOR VALUES ('LChenSecure','YU CHEN','FATHER');

INSERT INTO SPONSOR VALUES ('AMohamedSecure','ABU MOHAMED','BROTHER');

INSERT INTO SPONSOR VALUES ('TSchmidtSecure','LISA SCHMIDT','MOTHER');


INSERT INTO LOAN VALUES ('RsharmaSecure','SBI','6MONTHS');

INSERT INTO LOAN VALUES ('ViratKohliSecure','HDFC','6MONTHS');

INSERT INTO LOAN VALUES ('KYadavSecure','DCB','1YR');

INSERT INTO LOAN VALUES ('LChenSecure','SBI','6MONTHS');

INSERT INTO LOAN VALUES ('AMohamedSecure','HDFC','1YR');

INSERT INTO LOAN VALUES ('TSchmidtSecure','SBI','6MONTHS');

INSERT INTO LOAN VALUES ('YJaiswalSecure','DCB','1YR');
```

SCRIPT 2 : SQL QUERIES

```
----------1. Statement using Projection arithmetic (expressions)
SELECT
    Passport_ID,
    Visa_Type,
    First_Name,
    Last_Name,
    Date_of_Birth,
    Email_ID,
    Phone_Number,
    Nationality,
    (Phone_Number / 2) AS Half_Phone_Number
FROM Visa;


----2. Statement using Alias
SELECT
    v.Passport_ID AS ID,
    v.First_Name || ' ' || v.Last_Name AS Full_Name,
    t.Duration_Of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;


----3. Statement using Concatenation
SELECT
    First_Name || ' ' || Last_Name AS Full_Name
FROM Visa;


----4. Statement using The WHERE Clause
SELECT *
FROM Visa
```

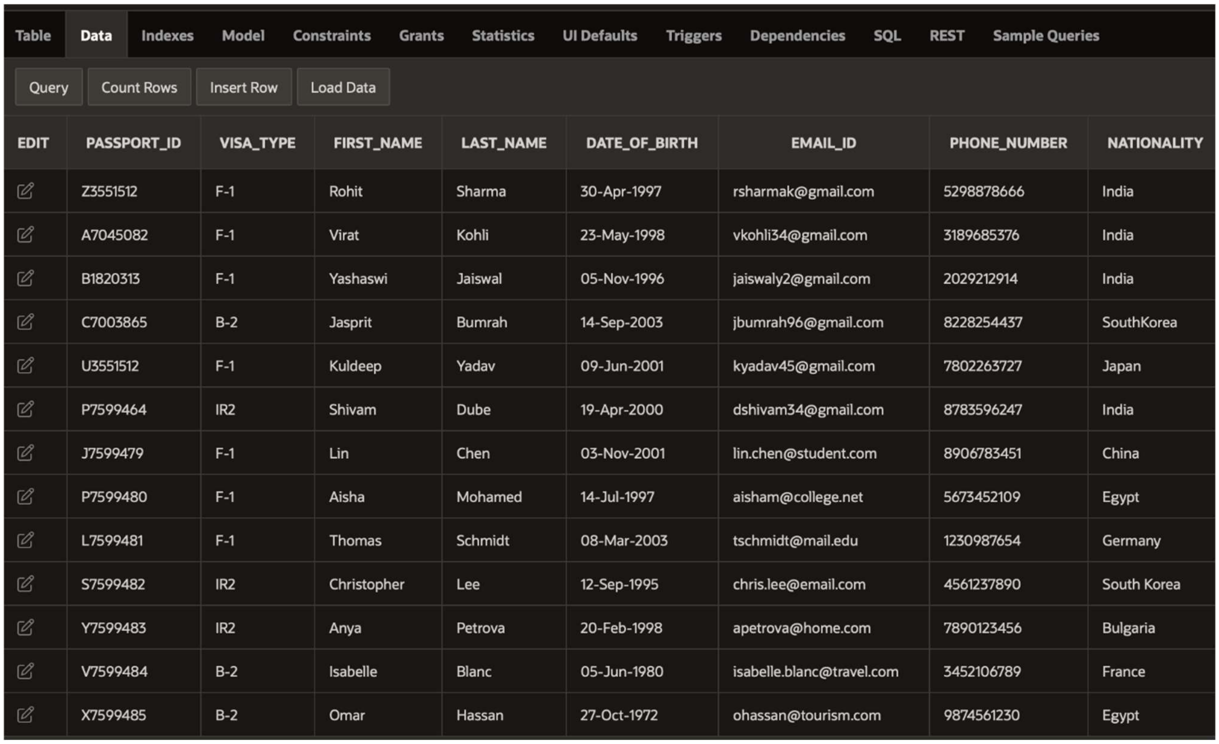
```sql
WHERE Nationality = 'India';

----5. Statement using Comparison Operators (Greater than...)
SELECT *
FROM Student
WHERE Student_ID > 50000000;

----6. Statement using Logical Operators (AND |OR
SELECT *
FROM Student s
LEFT JOIN FinancialAid fa ON s.Student_ID = fa.Student_ID
LEFT JOIN Scholarship sch ON fa.FinancialAidID = sch.FinancialAidID
LEFT JOIN Employment emp ON s.Student_ID = emp.Student_ID
WHERE (sch.CGPA > 3.0 OR emp.Department = 'Dining');

----7. Statement using Date Functions
SELECT *
FROM Student
WHERE Visa_Expiration_Date > SYSDATE;

----8. Statement using IN & subquery
SELECT *
FROM Visa
WHERE Nationality IN (SELECT Nationality FROM Visa WHERE Nationality = 'India' OR Nationality = 'Egypt');

-----9. Statement using LIKE OPERATOR
SELECT *
FROM Visa
WHERE Email_ID LIKE '%.com';

----10. Statement using AGGREGATION FUNCTIONS & GROUP BY
```

```
SELECT Section.Course_ID, Course.Course_Name, COUNT(Enrollment.Student_ID) AS Total_Students
FROM Section
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
JOIN Course ON Section.Course_ID = Course.Course_ID
GROUP BY Section.Course_ID, Course.Course_Name;

----11. Statement using GROUP BY & HAVING
SELECT Course.Course_ID, Course.Course_Name, COUNT(Enrollment.Enrollment_ID) AS Enrollment_Count
FROM Course
JOIN Section ON Course.Course_ID = Section.Course_ID
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
GROUP BY Course.Course_ID, Course.Course_Name
HAVING COUNT(Enrollment.Enrollment_ID) >= 2;

----12. Statement using Conditional Expressions
SELECT Passport_ID,
    First_Name,
    Last_Name,
    CASE
      WHEN Visa_Type = 'F-1' THEN 'Student Visa'
      WHEN Visa_Type = 'B-2' THEN 'Tourist Visa'
      ELSE 'Other'
    END AS Visa_Category
FROM Visa;

----13. Statement using "Union" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
UNION
SELECT Passport_ID FROM Tourist;

SELECT DISTINCT v.Passport_ID
```

```sql
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----14. Statement using "Minus" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
MINUS
SELECT Passport_ID FROM Tourist;

SELECT v.Passport_ID
FROM Visa v
LEFT JOIN Tourist t ON v.Passport_ID = t.Passport_ID
WHERE t.Passport_ID IS NULL;

----15. Statement using "INTERSECT" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
INTERSECT
SELECT Passport_ID FROM Tourist;

SELECT v.Passport_ID
FROM Visa v
INNER JOIN Tourist t ON v.Passport_ID = t.Passport_ID;
```

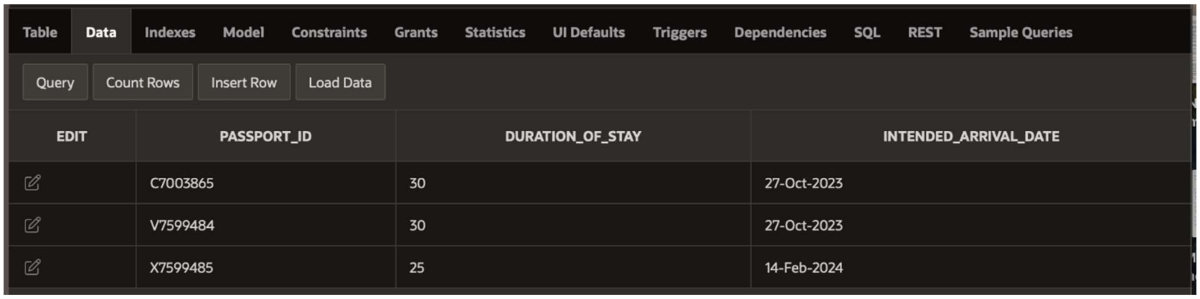
Tables:

VISA:



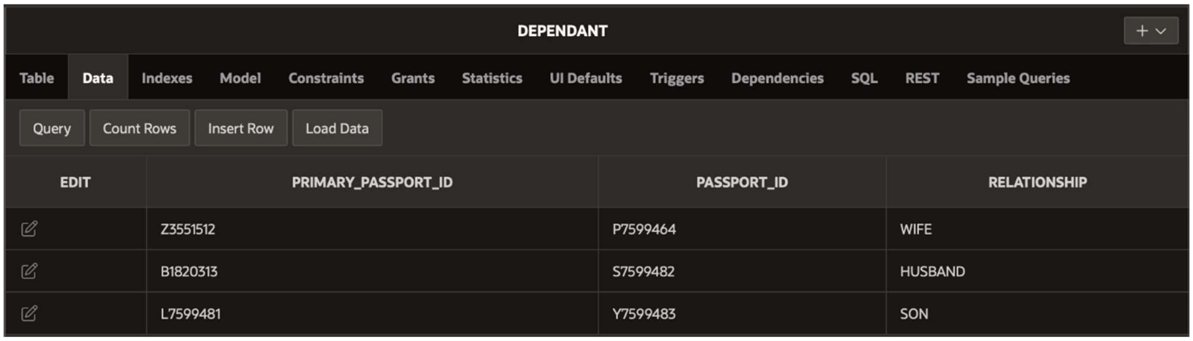
EDIT	PASSPORT_ID	VISA_TYPE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL_ID	PHONE_NUMBER	NATIONALITY
✎	Z3551512	F-1	Rohit	Sharma	30-Apr-1997	rsharmak@gmail.com	5298878666	India
✎	A7045082	F-1	Virat	Kohli	23-May-1998	vkohli34@gmail.com	3189685376	India
✎	B1820313	F-1	Yashaswi	Jaiswal	05-Nov-1996	jaiswaly2@gmail.com	2029212914	India
✎	C7003865	B-2	Jasprit	Bumrah	14-Sep-2003	jbumrah96@gmail.com	8228254437	SouthKorea
✎	U3551512	F-1	Kuldeep	Yadav	09-Jun-2001	kyadav45@gmail.com	7802263727	Japan
✎	P7599464	IR2	Shivam	Dube	19-Apr-2000	dshivam34@gmail.com	8783596247	India
✎	J7599479	F-1	Lin	Chen	03-Nov-2001	lin.chen@student.com	8906783451	China
✎	P7599480	F-1	Aisha	Mohamed	14-Jul-1997	aisham@college.net	5673452109	Egypt
✎	L7599481	F-1	Thomas	Schmidt	08-Mar-2003	tschmidt@mail.edu	1230987654	Germany
✎	S7599482	IR2	Christopher	Lee	12-Sep-1995	chris.lee@email.com	4561237890	South Korea
✎	Y7599483	IR2	Anya	Petrova	20-Feb-1998	apetrova@home.com	7890123456	Bulgaria
✎	V7599484	B-2	Isabelle	Blanc	05-Jun-1980	isabelle.blanc@travel.com	3452106789	France
✎	X7599485	B-2	Omar	Hassan	27-Oct-1972	ohassan@tourism.com	9874561230	Egypt

TOURIST:



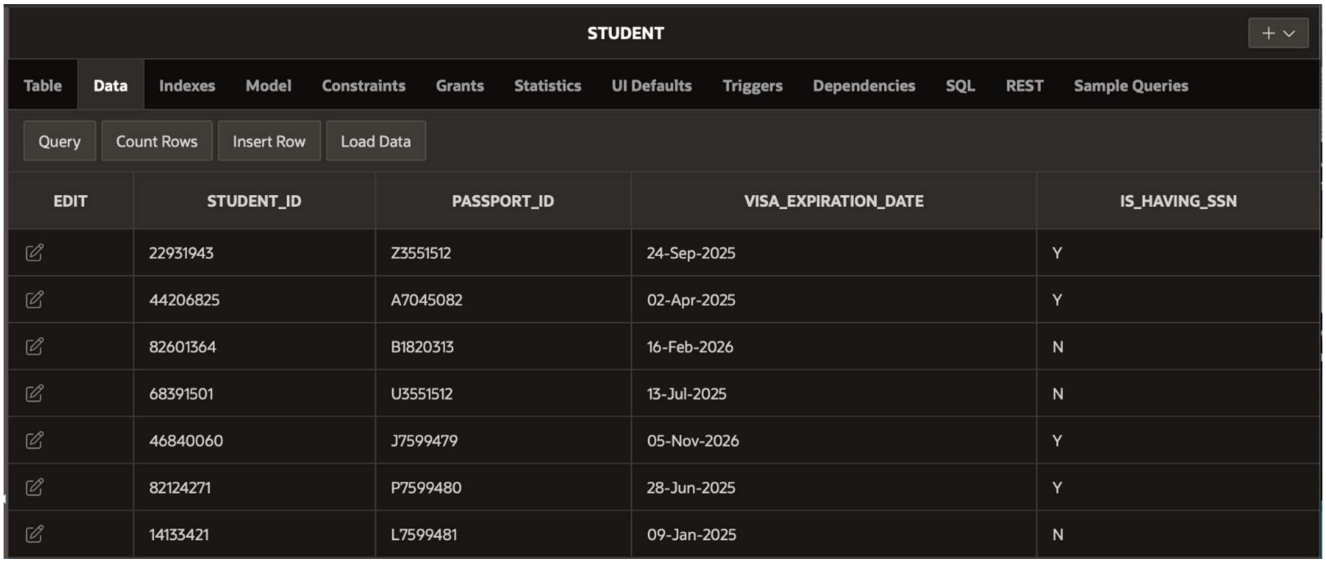
EDIT	PASSPORT_ID	DURATION_OF_STAY	INTENDED_ARRIVAL_DATE
✎	C7003865	30	27-Oct-2023
✎	V7599484	30	27-Oct-2023
✎	X7599485	25	14-Feb-2024

DEPENDANT:



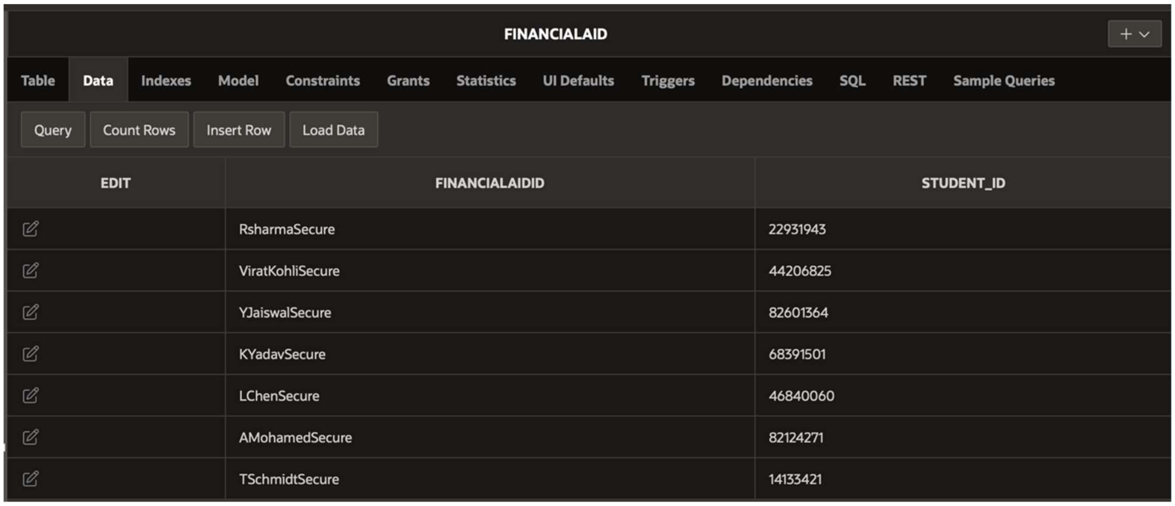
EDIT	PRIMARY_PASSPORT_ID	PASSPORT_ID	RELATIONSHIP
✎	Z3551512	P7599464	WIFE
✎	B1820313	S7599482	HUSBAND
✎	L7599481	Y7599483	SON

STUDENT:



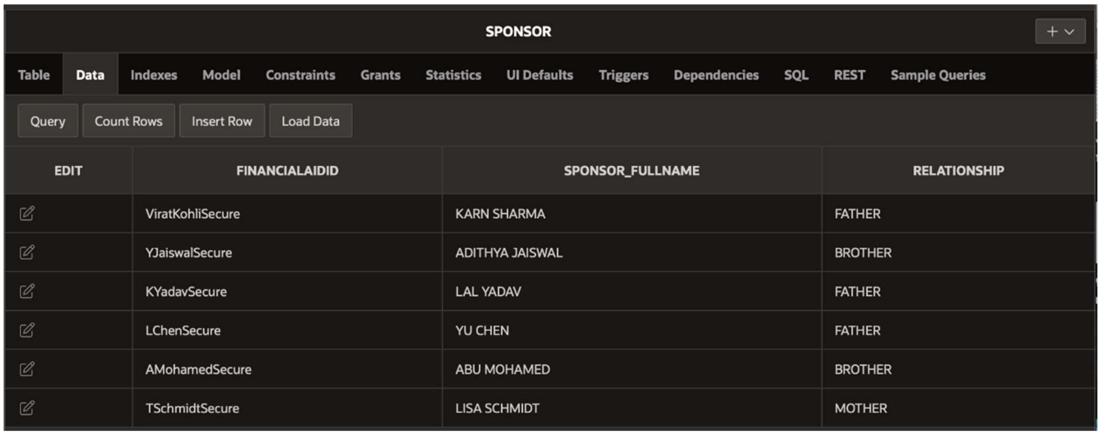
EDIT	STUDENT_ID	PASSPORT_ID	VISA_EXPIRATION_DATE	IS_HAVING_SSN
✎	22931943	Z3551512	24-Sep-2025	Y
✎	44206825	A7045082	02-Apr-2025	Y
✎	82601364	B1820313	16-Feb-2026	N
✎	68391501	U3551512	13-Jul-2025	N
✎	46840060	J7599479	05-Nov-2026	Y
✎	82124271	P7599480	28-Jun-2025	Y
✎	14133421	L7599481	09-Jan-2025	N

FINANCIALAID:



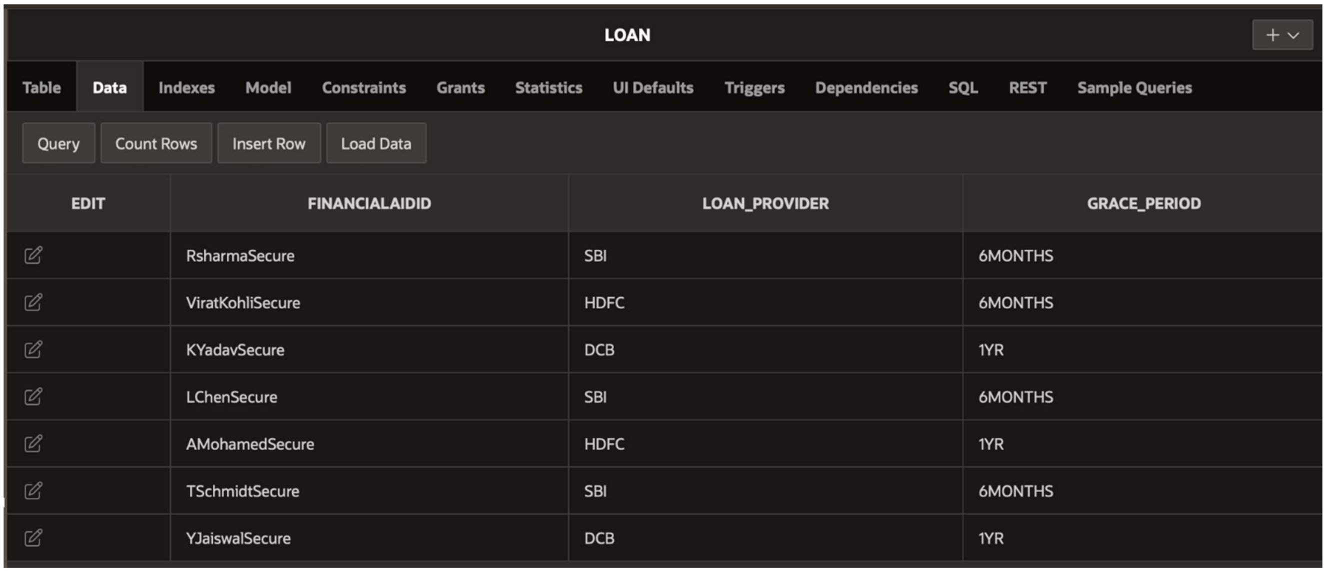
EDIT	FINANCIALAIDID	STUDENT_ID
✎	RsharmaSecure	22931943
✎	ViratKohliSecure	44206825
✎	YJaiswalSecure	82601364
✎	KYadavSecure	68391501
✎	LChenSecure	46840060
✎	AMohamedSecure	82124271
✎	TSchmidtSecure	14133421

SPONSOR:



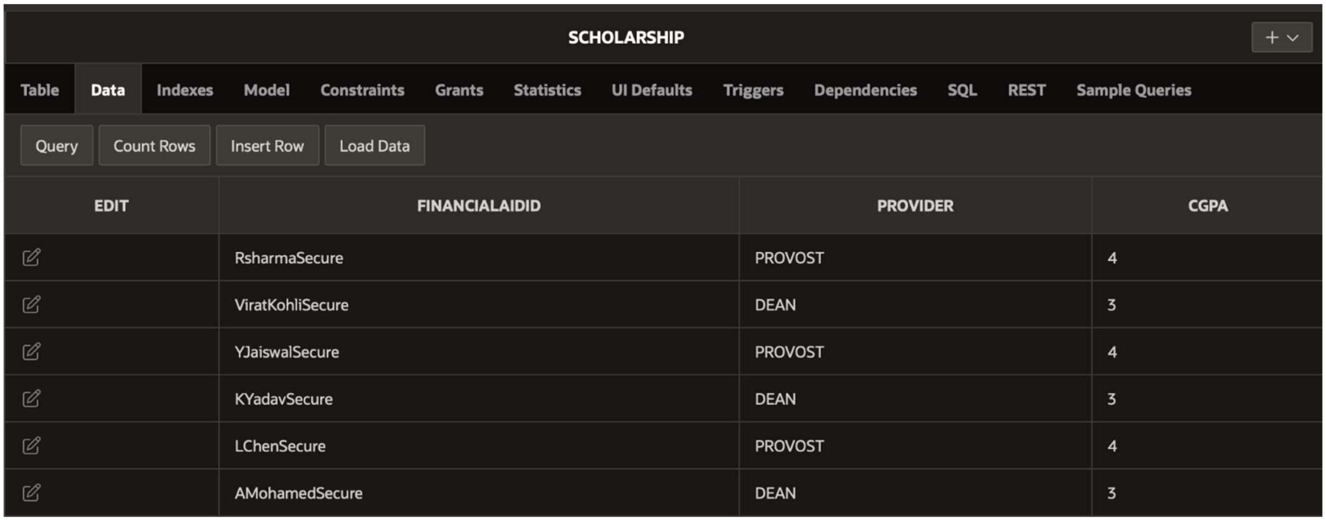
EDIT	FINANCIALAIDID	SPONSOR_FULLNAME	RELATIONSHIP
✎	ViratKohliSecure	KARN SHARMA	FATHER
✎	YJaiswalSecure	ADITHYA JAISWAL	BROTHER
✎	KYadavSecure	LAL YADAV	FATHER
✎	LChenSecure	YU CHEN	FATHER
✎	AMohamedSecure	ABU MOHAMED	BROTHER
✎	TSchmidtSecure	LISA SCHMIDT	MOTHER

LOAN:



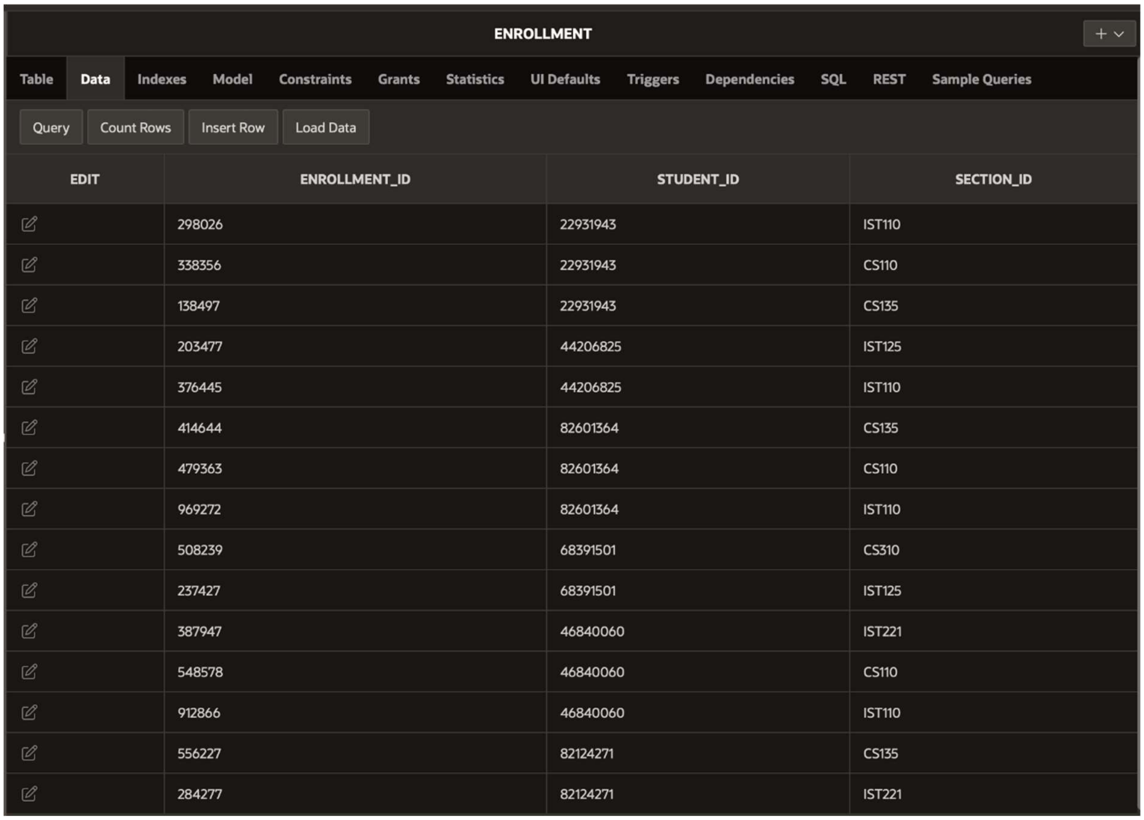
EDIT	FINANCIALAIDID	LOAN_PROVIDER	GRACE_PERIOD
✎	RsharmaSecure	SBI	6MONTHS
✎	ViratKohliSecure	HDFC	6MONTHS
✎	KYadavSecure	DCB	1YR
✎	LChenSecure	SBI	6MONTHS
✎	AMohamedSecure	HDFC	1YR
✎	TSchmidtSecure	SBI	6MONTHS
✎	YJaiswalSecure	DCB	1YR

SCHOLARSHIP:



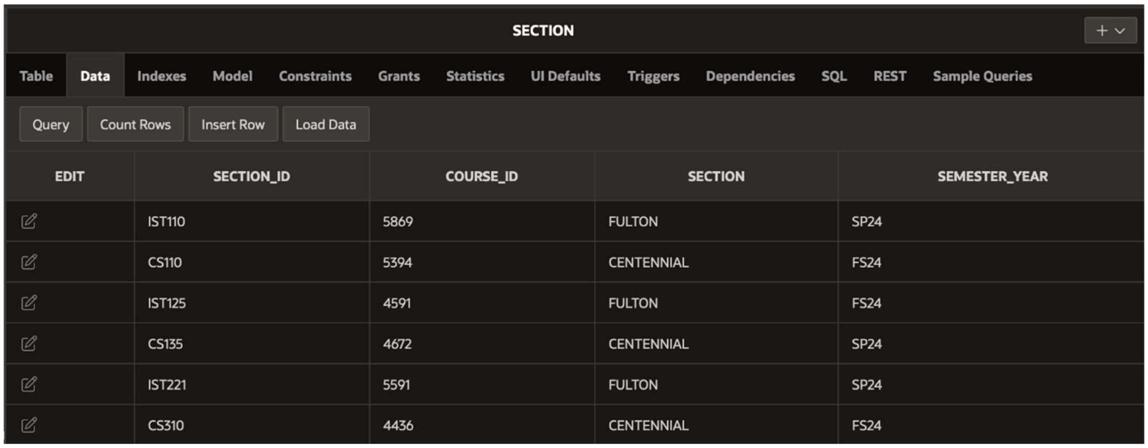
EDIT	FINANCIALAIDID	PROVIDER	CGPA
✎	RsharmaSecure	PROVOST	4
✎	ViratKohliSecure	DEAN	3
✎	YJaiswalSecure	PROVOST	4
✎	KYadavSecure	DEAN	3
✎	LChenSecure	PROVOST	4
✎	AMohamedSecure	DEAN	3

ENROLLMENT:



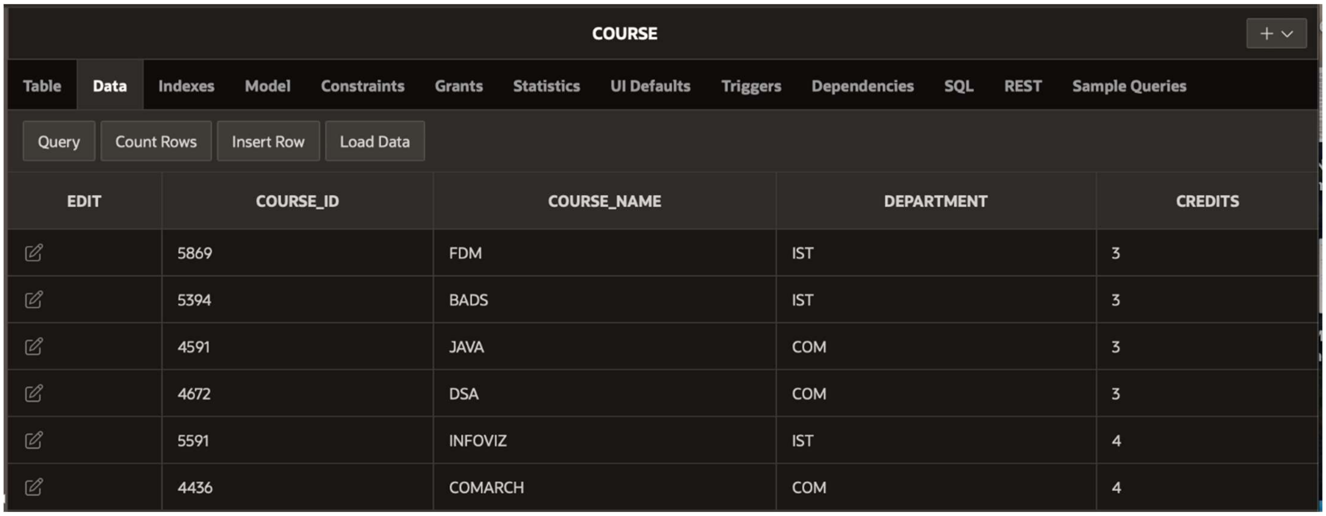
EDIT	ENROLLMENT_ID	STUDENT_ID	SECTION_ID
✎	298026	22931943	IST110
✎	338356	22931943	CS110
✎	138497	22931943	CS135
✎	203477	44206825	IST125
✎	376445	44206825	IST110
✎	414644	82601364	CS135
✎	479363	82601364	CS110
✎	969272	82601364	IST110
✎	508239	68391501	CS310
✎	237427	68391501	IST125
✎	387947	46840060	IST221
✎	548578	46840060	CS110
✎	912866	46840060	IST110
✎	556227	82124271	CS135
✎	284277	82124271	IST221

SECTION:



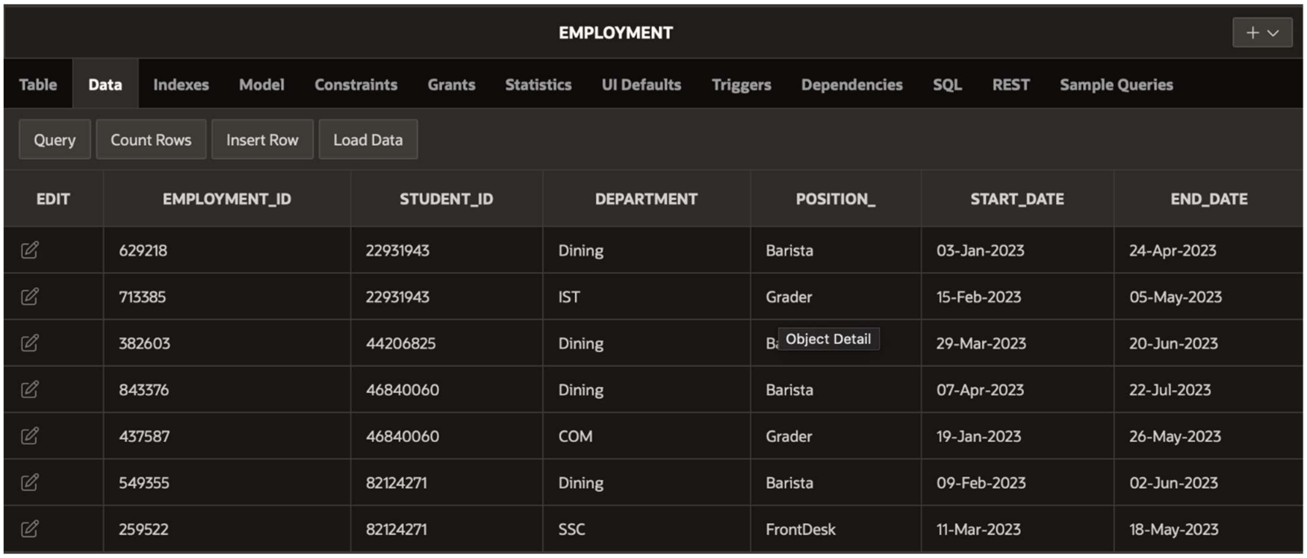
EDIT	SECTION_ID	COURSE_ID	SECTION	SEMESTER_YEAR
✎	IST110	5869	FULTON	SP24
✎	CS110	5394	CENTENNIAL	FS24
✎	IST125	4591	FULTON	FS24
✎	CS135	4672	CENTENNIAL	SP24
✎	IST221	5591	FULTON	SP24
✎	CS310	4436	CENTENNIAL	FS24

COURSE:



EDIT	COURSE_ID	COURSE_NAME	DEPARTMENT	CREDITS
✎	5869	FDM	IST	3
✎	5394	BADS	IST	3
✎	4591	JAVA	COM	3
✎	4672	DSA	COM	3
✎	5591	INFOVIZ	IST	4
✎	4436	COMARCH	COM	4

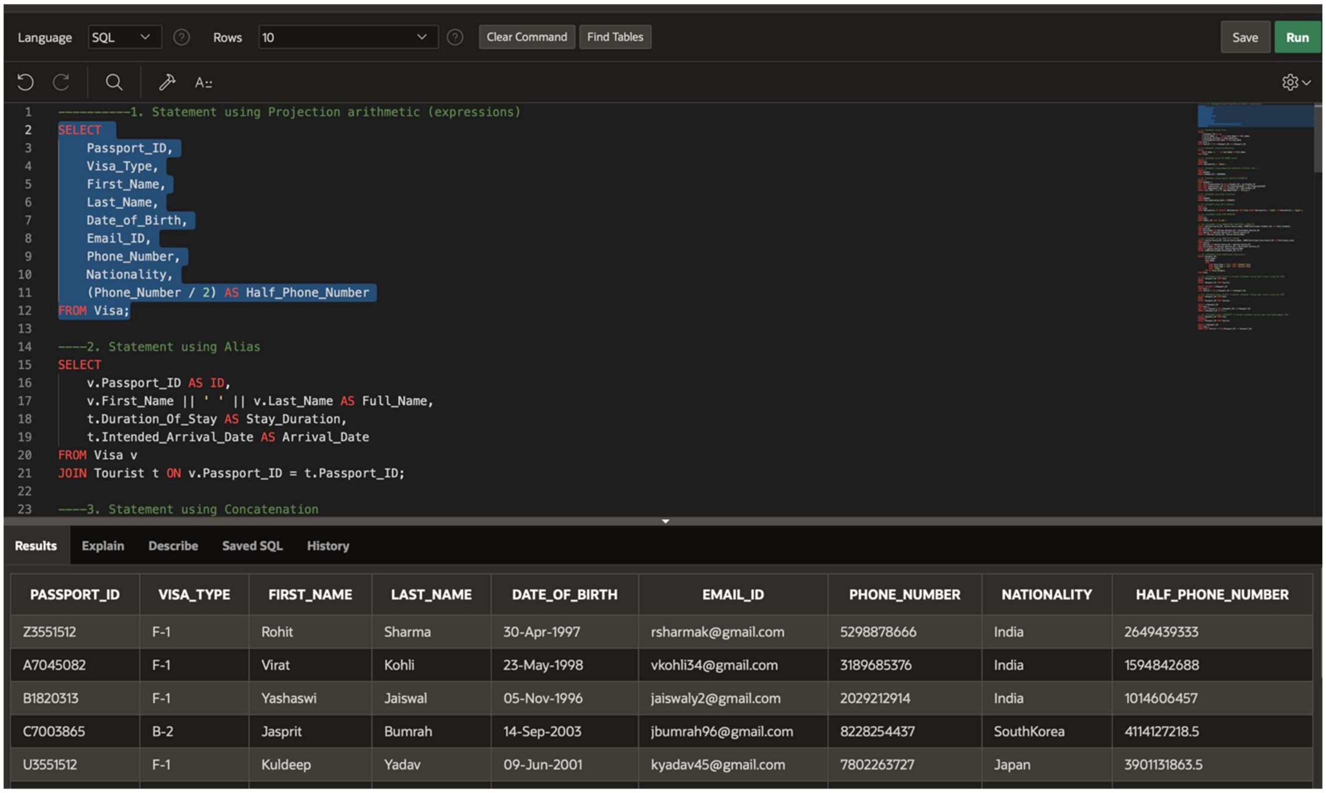
EMPLOYMENT:



EDIT	EMPLOYMENT_ID	STUDENT_ID	DEPARTMENT	POSITION_	START_DATE	END_DATE
✎	629218	22931943	Dining	Barista	03-Jan-2023	24-Apr-2023
✎	713385	22931943	IST	Grader	15-Feb-2023	05-May-2023
✎	382603	44206825	Dining	Ba	29-Mar-2023	20-Jun-2023
✎	843376	46840060	Dining	Barista	07-Apr-2023	22-Jul-2023
✎	437587	46840060	COM	Grader	19-Jan-2023	26-May-2023
✎	549355	82124271	Dining	Barista	09-Feb-2023	02-Jun-2023
✎	259522	82124271	SSC	FrontDesk	11-Mar-2023	18-May-2023

SCREENSHOTS QUERIES:

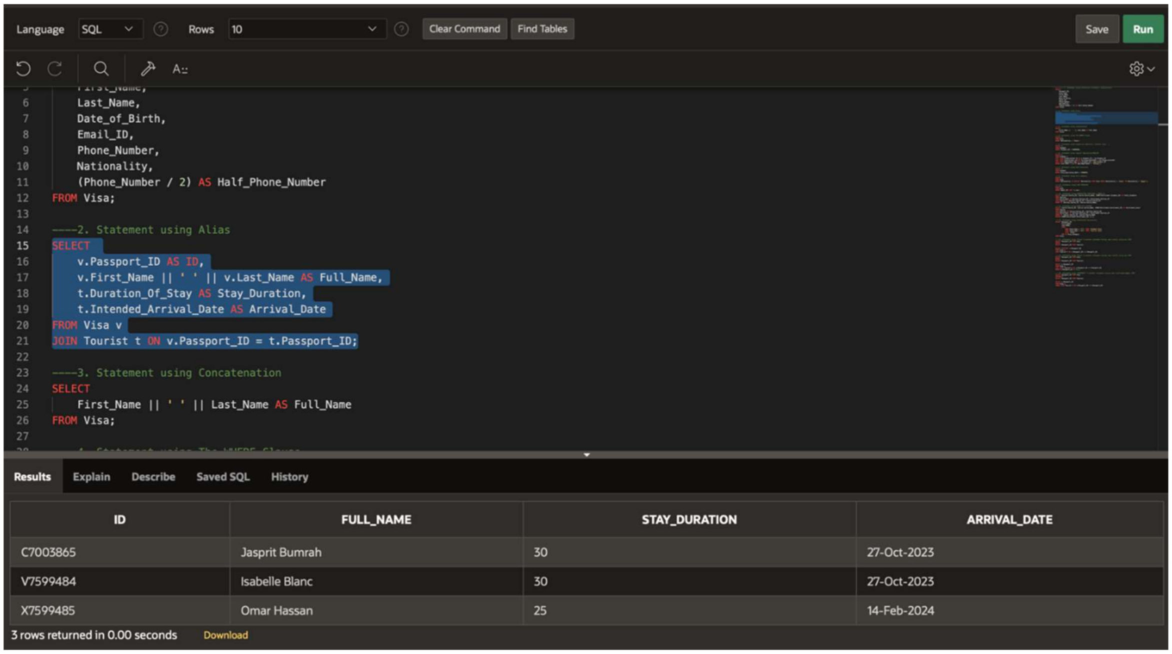
1.



```
----------1. Statement using Projection arithmetic (expressions)
SELECT
    Passport_ID,
    Visa_Type,
    First_Name,
    Last_Name,
    Date_of_Birth,
    Email_ID,
    Phone_Number,
    Nationality,
    (Phone_Number / 2) AS Half_Phone_Number
FROM Visa;

----2. Statement using Alias
SELECT
    v.Passport_ID AS ID,
    v.First_Name || ' ' || v.Last_Name AS Full_Name,
    t.Duration_Of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----3. Statement using Concatenation
```

PASSPORT_ID	VISA_TYPE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL_ID	PHONE_NUMBER	NATIONALITY	HALF_PHONE_NUMBER
Z3551512	F-1	Rohit	Sharma	30-Apr-1997	rsharmak@gmail.com	5298878666	India	2649439333
A7045082	F-1	Virat	Kohli	23-May-1998	vkohli34@gmail.com	3189685376	India	1594842688
B1820313	F-1	Yashaswi	Jaiswal	05-Nov-1996	jaiswaly2@gmail.com	2029212914	India	1014606457
C7003865	B-2	Jasprit	Bumrah	14-Sep-2003	jbumrah96@gmail.com	8228254437	SouthKorea	4114127218.5
U3551512	F-1	Kuldeep	Yadav	09-Jun-2001	kyadav45@gmail.com	7802263727	Japan	3901131863.5

2.

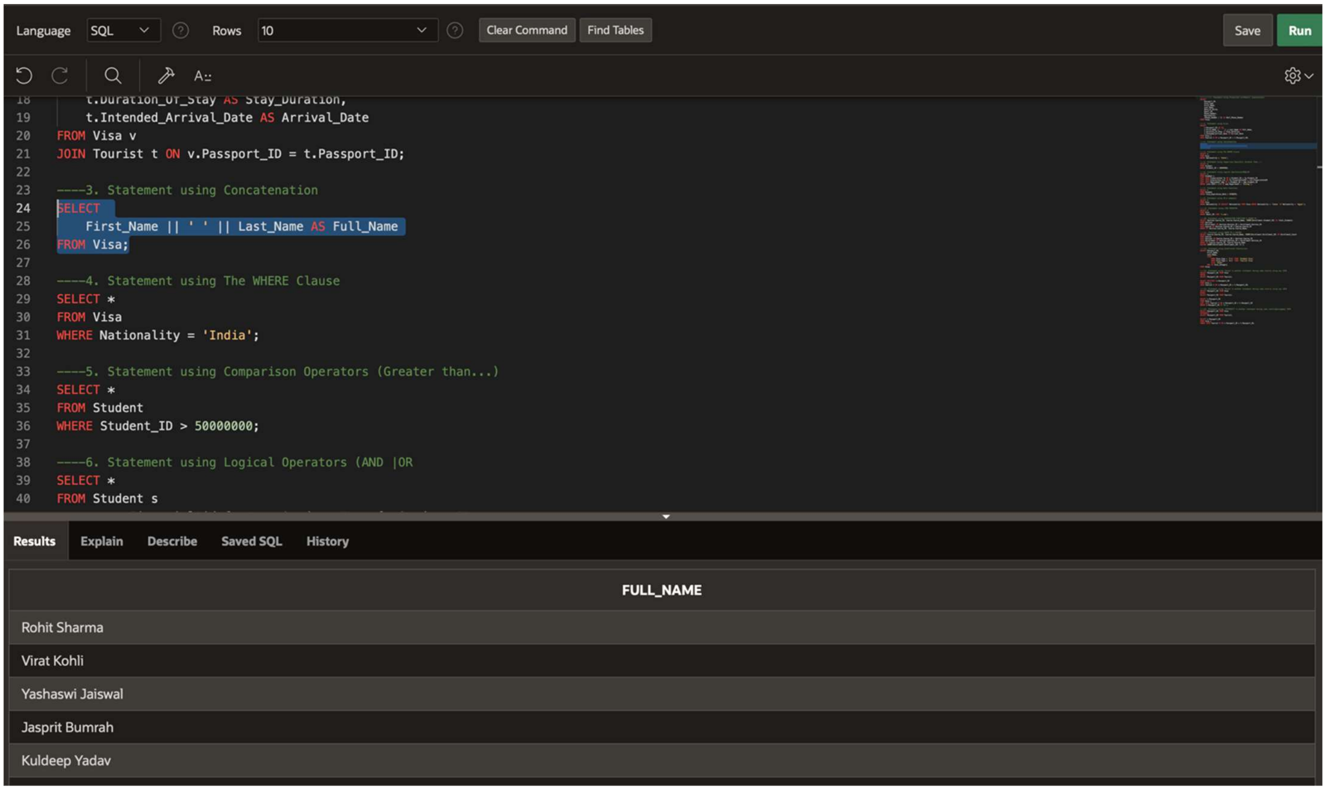


```
    First_Name,
    Last_Name,
    Date_of_Birth,
    Email_ID,
    Phone_Number,
    Nationality,
    (Phone_Number / 2) AS Half_Phone_Number
FROM Visa;

----2. Statement using Alias
SELECT
    v.Passport_ID AS ID,
    v.First_Name || ' ' || v.Last_Name AS Full_Name,
    t.Duration_Of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----3. Statement using Concatenation
SELECT
    First_Name || ' ' || Last_Name AS Full_Name
FROM Visa;

----4. Statement using The WHERE Clause
```

ID	FULL_NAME	STAY_DURATION	ARRIVAL_DATE
C7003865	Jasprit Bumrah	30	27-Oct-2023
V7599484	Isabelle Blanc	30	27-Oct-2023
X7599485	Omar Hassan	25	14-Feb-2024

3 rows returned in 0.00 seconds

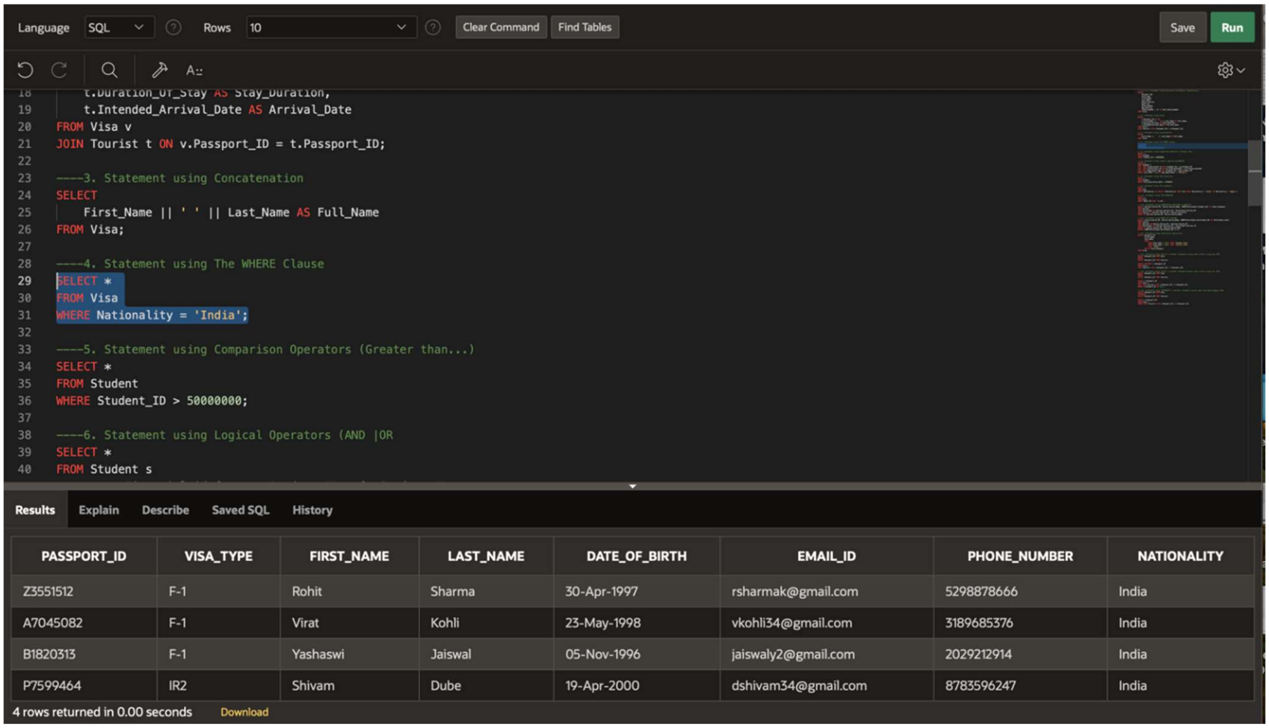
3.



```sql
    t.Duration_Of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----3. Statement using Concatenation
SELECT
    First_Name || ' ' || Last_Name AS Full_Name
FROM Visa;

----4. Statement using The WHERE Clause
SELECT *
FROM Visa
WHERE Nationality = 'India';

----5. Statement using Comparison Operators (Greater than...)
SELECT *
FROM Student
WHERE Student_ID > 50000000;

----6. Statement using Logical Operators (AND |OR
SELECT *
FROM Student s
```

FULL_NAME
Rohit Sharma
Virat Kohli
Yashaswi Jaiswal
Jasprit Bumrah
Kuldeep Yadav

4.

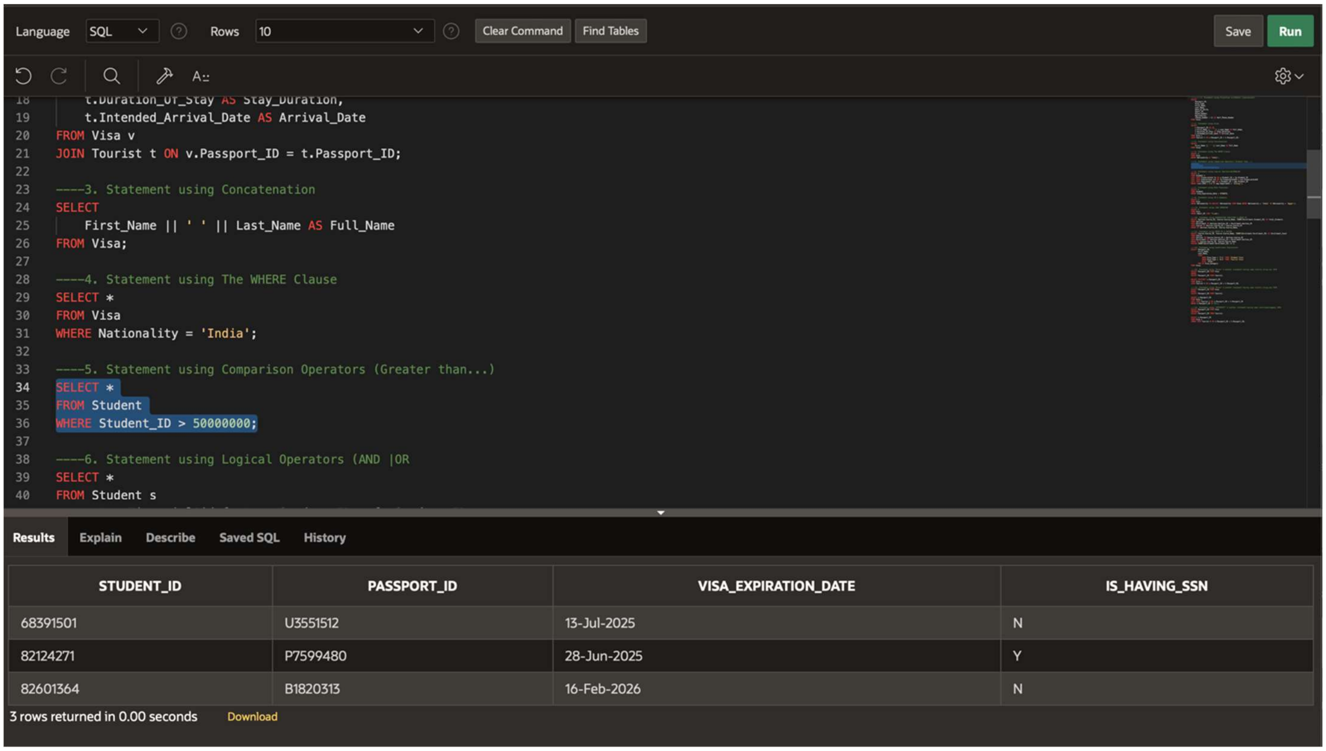


```sql
    t.Duration_Of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----3. Statement using Concatenation
SELECT
    First_Name || ' ' || Last_Name AS Full_Name
FROM Visa;

----4. Statement using The WHERE Clause
SELECT *
FROM Visa
WHERE Nationality = 'India';

----5. Statement using Comparison Operators (Greater than...)
SELECT *
FROM Student
WHERE Student_ID > 50000000;

----6. Statement using Logical Operators (AND |OR
SELECT *
FROM Student s
```

PASSPORT_ID	VISA_TYPE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL_ID	PHONE_NUMBER	NATIONALITY
Z3551512	F-1	Rohit	Sharma	30-Apr-1997	rsharmak@gmail.com	5298878666	India
A7045082	F-1	Virat	Kohli	23-May-1998	vkohli34@gmail.com	3189685376	India
B1820313	F-1	Yashaswi	Jaiswal	05-Nov-1996	jaiswaly2@gmail.com	2029212914	India
P7599464	IR2	Shivam	Dube	19-Apr-2000	dshivam34@gmail.com	8783596247	India

4 rows returned in 0.00 seconds

5.

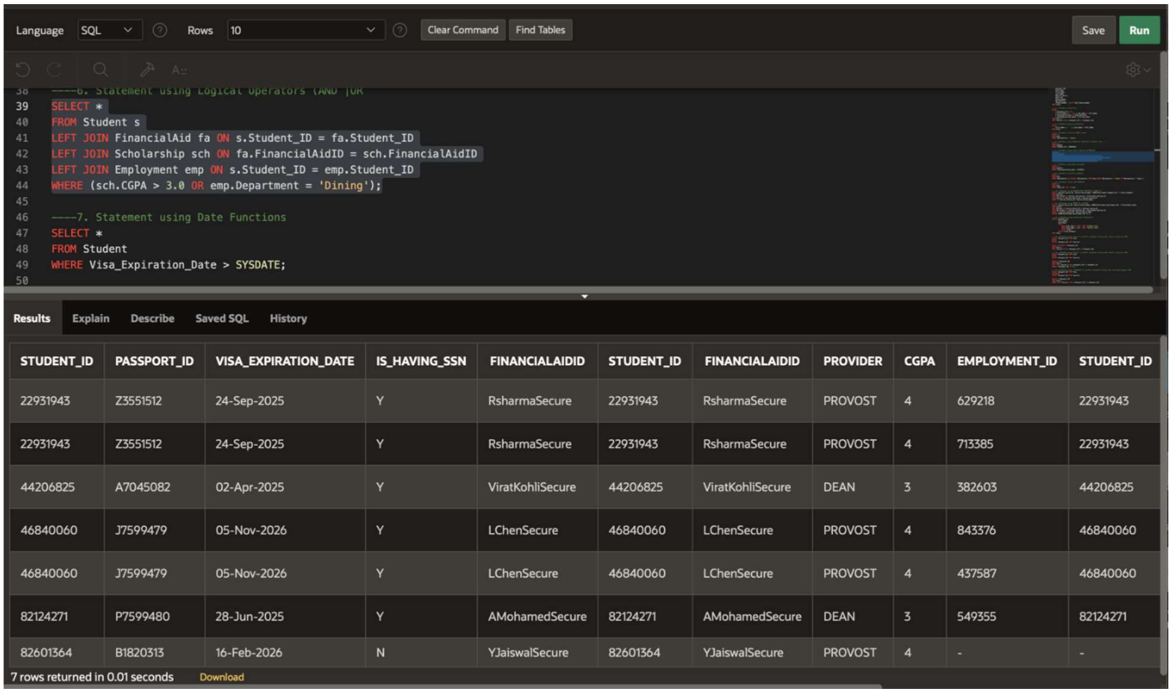


```
    t.Duration_of_Stay AS Stay_Duration,
    t.Intended_Arrival_Date AS Arrival_Date
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----3. Statement using Concatenation
SELECT
    First_Name || ' ' || Last_Name AS Full_Name
FROM Visa;

----4. Statement using The WHERE Clause
SELECT *
FROM Visa
WHERE Nationality = 'India';

----5. Statement using Comparison Operators (Greater than...)
SELECT *
FROM Student
WHERE Student_ID > 50000000;

----6. Statement using Logical Operators (AND |OR
SELECT *
FROM Student s
```

STUDENT_ID	PASSPORT_ID	VISA_EXPIRATION_DATE	IS_HAVING_SSN
68391501	U3551512	13-Jul-2025	N
82124271	P7599480	28-Jun-2025	Y
82601364	B1820313	16-Feb-2026	N

3 rows returned in 0.00 seconds

6.

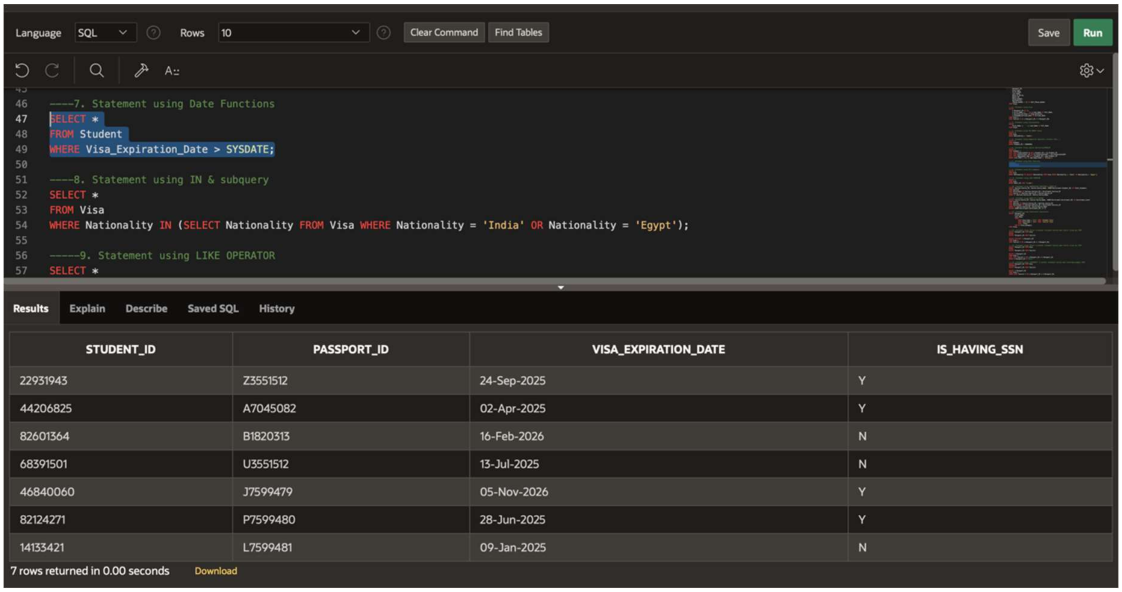


```
----6. Statement using Logical Operators (AND |OR
SELECT *
FROM Student s
LEFT JOIN FinancialAid fa ON s.Student_ID = fa.Student_ID
LEFT JOIN Scholarship sch ON fa.FinancialAidID = sch.FinancialAidID
LEFT JOIN Employment emp ON s.Student_ID = emp.Student_ID
WHERE (sch.CGPA > 3.0 OR emp.Department = 'Dining');

----7. Statement using Date Functions
SELECT *
FROM Student
WHERE Visa_Expiration_Date > SYSDATE;
```

STUDENT_ID	PASSPORT_ID	VISA_EXPIRATION_DATE	IS_HAVING_SSN	FINANCIALAIDID	STUDENT_ID	FINANCIALAIDID	PROVIDER	CGPA	EMPLOYMENT_ID	STUDENT_ID
22931943	Z3551512	24-Sep-2025	Y	RsharmaSecure	22931943	RsharmaSecure	PROVOST	4	629218	22931943
22931943	Z3551512	24-Sep-2025	Y	RsharmaSecure	22931943	RsharmaSecure	PROVOST	4	713385	22931943
44206825	A7045082	02-Apr-2025	Y	ViratKohliSecure	44206825	ViratKohliSecure	DEAN	3	382603	44206825
46840060	J7599479	05-Nov-2026	Y	LChenSecure	46840060	LChenSecure	PROVOST	4	843376	46840060
46840060	J7599479	05-Nov-2026	Y	LChenSecure	46840060	LChenSecure	PROVOST	4	437587	46840060
82124271	P7599480	28-Jun-2025	Y	AMohamedSecure	82124271	AMohamedSecure	DEAN	3	549355	82124271
82601364	B1820313	16-Feb-2026	N	YJaiswalSecure	82601364	YJaiswalSecure	PROVOST	4	-	-

7 rows returned in 0.01 seconds

7.

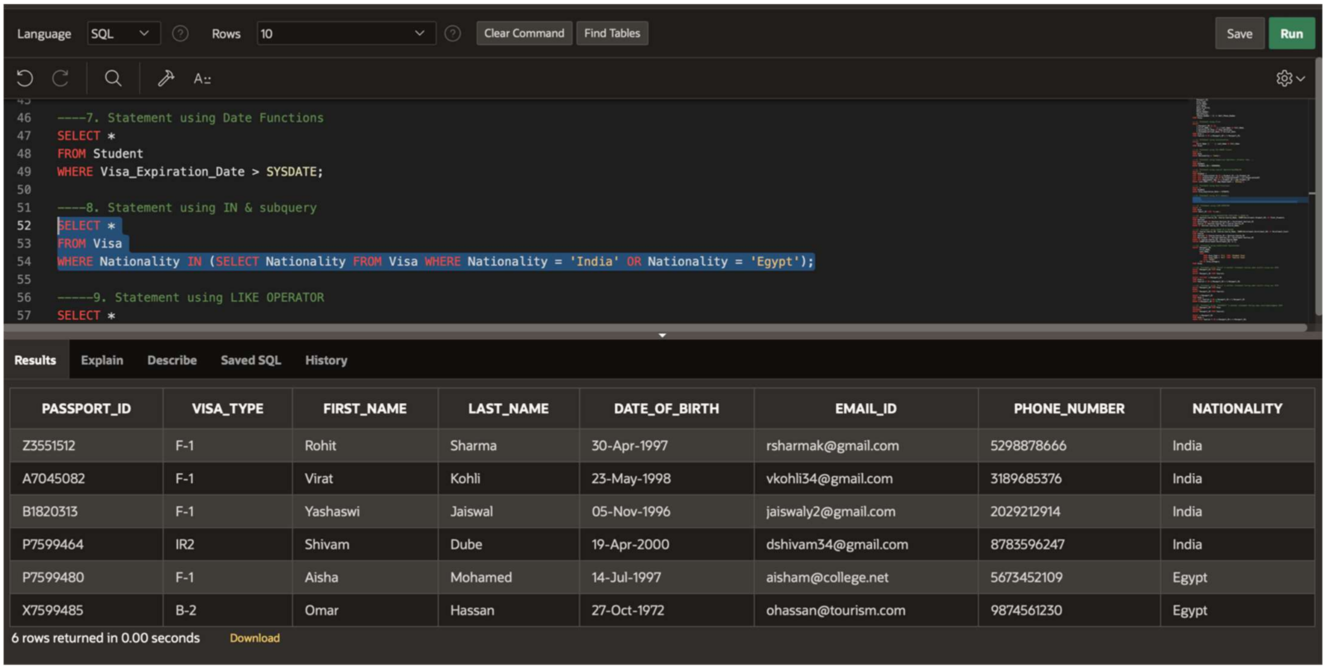


```
----7. Statement using Date Functions
SELECT *
FROM Student
WHERE Visa_Expiration_Date > SYSDATE;

----8. Statement using IN & subquery
SELECT *
FROM Visa
WHERE Nationality IN (SELECT Nationality FROM Visa WHERE Nationality = 'India' OR Nationality = 'Egypt');

-----9. Statement using LIKE OPERATOR
SELECT *
```

STUDENT_ID	PASSPORT_ID	VISA_EXPIRATION_DATE	IS_HAVING_SSN
22931943	Z3551512	24-Sep-2025	Y
44206825	A7045082	02-Apr-2025	Y
82601364	B1820313	16-Feb-2026	N
68391501	U3551512	13-Jul-2025	N
46840060	J7599479	05-Nov-2026	Y
82124271	P7599480	28-Jun-2025	Y
14133421	L7599481	09-Jan-2025	N

7 rows returned in 0.00 seconds

8.

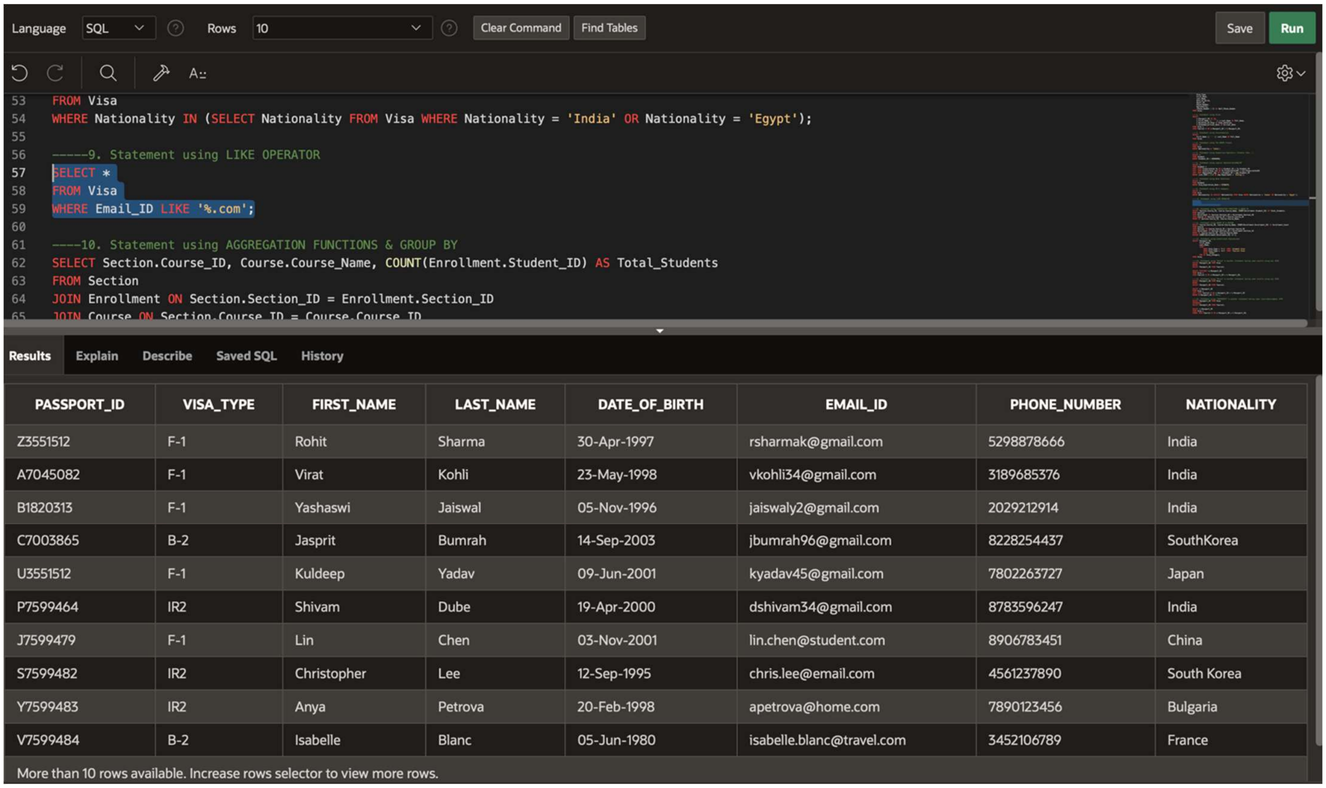


```
----7. Statement using Date Functions
SELECT *
FROM Student
WHERE Visa_Expiration_Date > SYSDATE;

----8. Statement using IN & subquery
SELECT *
FROM Visa
WHERE Nationality IN (SELECT Nationality FROM Visa WHERE Nationality = 'India' OR Nationality = 'Egypt');

-----9. Statement using LIKE OPERATOR
SELECT *
```

PASSPORT_ID	VISA_TYPE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL_ID	PHONE_NUMBER	NATIONALITY
Z3551512	F-1	Rohit	Sharma	30-Apr-1997	rsharmak@gmail.com	5298878666	India
A7045082	F-1	Virat	Kohli	23-May-1998	vkohli34@gmail.com	3189685376	India
B1820313	F-1	Yashaswi	Jaiswal	05-Nov-1996	jaiswaly2@gmail.com	2029212914	India
P7599464	IR2	Shivam	Dube	19-Apr-2000	dshivam34@gmail.com	8783596247	India
P7599480	F-1	Aisha	Mohamed	14-Jul-1997	aisham@college.net	5673452109	Egypt
X7599485	B-2	Omar	Hassan	27-Oct-1972	ohassan@tourism.com	9874561230	Egypt

6 rows returned in 0.00 seconds

9.



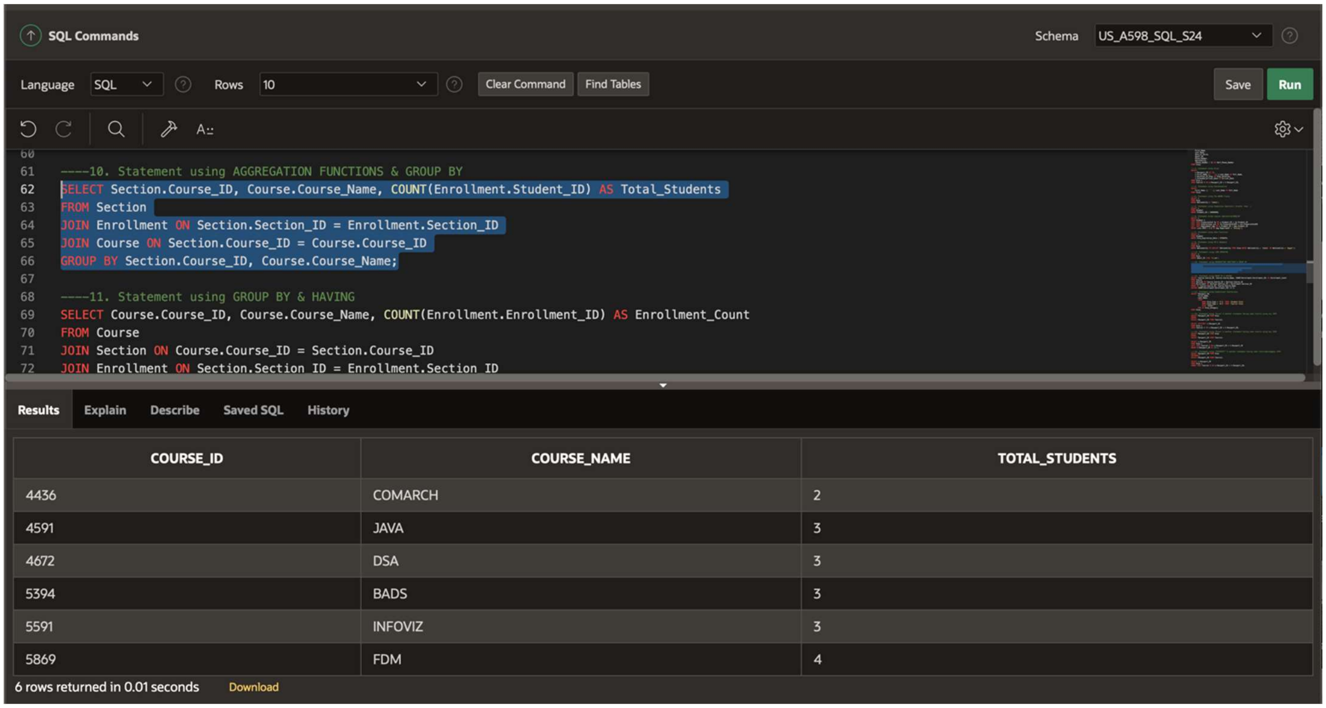
Language SQL   Rows 10

```
FROM Visa
WHERE Nationality IN (SELECT Nationality FROM Visa WHERE Nationality = 'India' OR Nationality = 'Egypt');

-----9. Statement using LIKE OPERATOR
SELECT *
FROM Visa
WHERE Email_ID LIKE '%.com';

----10. Statement using AGGREGATION FUNCTIONS & GROUP BY
SELECT Section.Course_ID, Course.Course_Name, COUNT(Enrollment.Student_ID) AS Total_Students
FROM Section
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
JOIN Course ON Section.Course_ID = Course.Course_ID
```

PASSPORT_ID	VISA_TYPE	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	EMAIL_ID	PHONE_NUMBER	NATIONALITY
Z3551512	F-1	Rohit	Sharma	30-Apr-1997	rsharmak@gmail.com	5298878666	India
A7045082	F-1	Virat	Kohli	23-May-1998	vkohli34@gmail.com	3189685376	India
B1820313	F-1	Yashaswi	Jaiswal	05-Nov-1996	jaiswaly2@gmail.com	2029212914	India
C7003865	B-2	Jasprit	Bumrah	14-Sep-2003	jbumrah96@gmail.com	8228254437	SouthKorea
U3551512	F-1	Kuldeep	Yadav	09-Jun-2001	kyadav45@gmail.com	7802263727	Japan
P7599464	IR2	Shivam	Dube	19-Apr-2000	dshivam34@gmail.com	8783596247	India
J7599479	F-1	Lin	Chen	03-Nov-2001	lin.chen@student.com	8906783451	China
S7599482	IR2	Christopher	Lee	12-Sep-1995	chris.lee@email.com	4561237890	South Korea
Y7599483	IR2	Anya	Petrova	20-Feb-1998	apetrova@home.com	7890123456	Bulgaria
V7599484	B-2	Isabelle	Blanc	05-Jun-1980	isabelle.blanc@travel.com	3452106789	France

More than 10 rows available. Increase rows selector to view more rows.

10.



Schema US_A598_SQL_S24

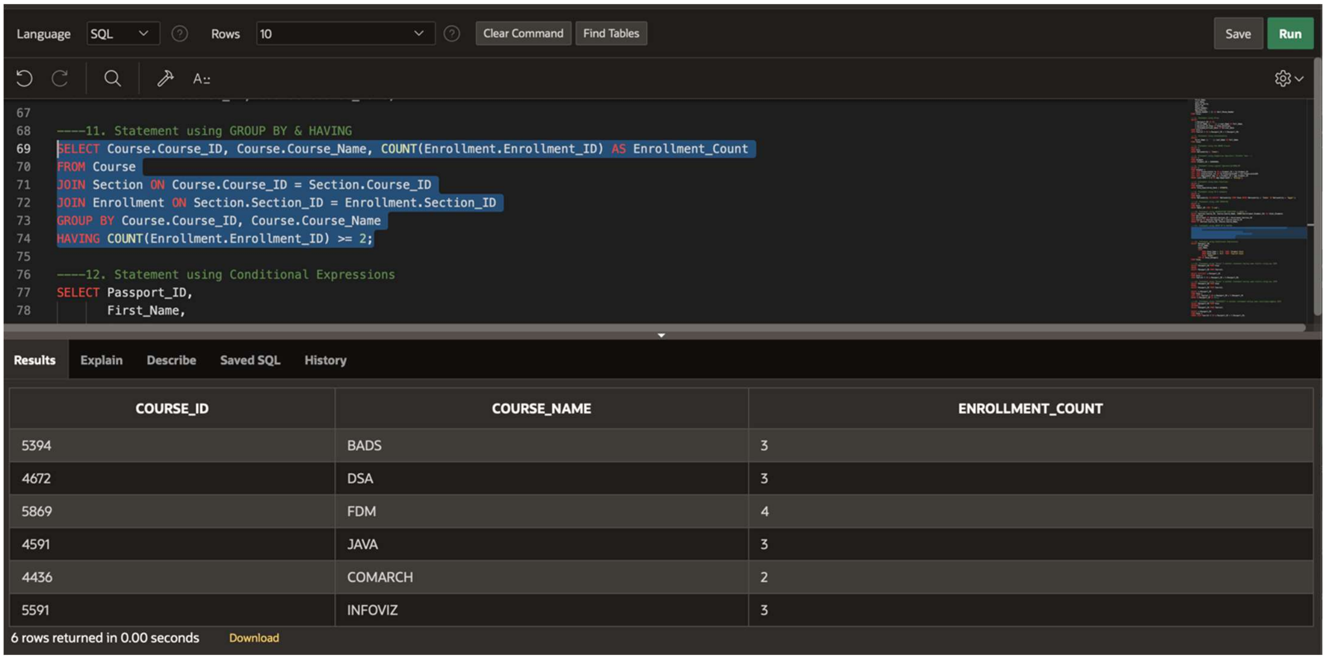
Language SQL   Rows 10

```
----10. Statement using AGGREGATION FUNCTIONS & GROUP BY
SELECT Section.Course_ID, Course.Course_Name, COUNT(Enrollment.Student_ID) AS Total_Students
FROM Section
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
JOIN Course ON Section.Course_ID = Course.Course_ID
GROUP BY Section.Course_ID, Course.Course_Name;

----11. Statement using GROUP BY & HAVING
SELECT Course.Course_ID, Course.Course_Name, COUNT(Enrollment.Enrollment_ID) AS Enrollment_Count
FROM Course
JOIN Section ON Course.Course_ID = Section.Course_ID
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
```

COURSE_ID	COURSE_NAME	TOTAL_STUDENTS
4436	COMARCH	2
4591	JAVA	3
4672	DSA	3
5394	BADS	3
5591	INFOVIZ	3
5869	FDM	4

6 rows returned in 0.01 seconds

11.

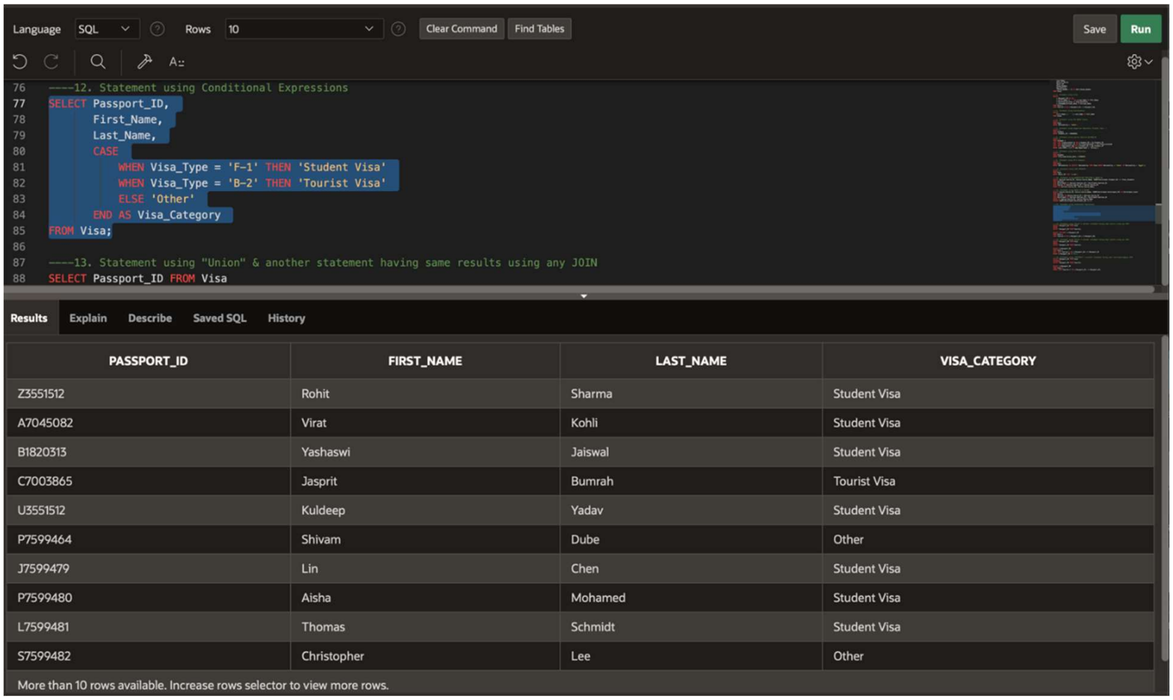


```
----11. Statement using GROUP BY & HAVING
SELECT Course.Course_ID, Course.Course_Name, COUNT(Enrollment.Enrollment_ID) AS Enrollment_Count
FROM Course
JOIN Section ON Course.Course_ID = Section.Course_ID
JOIN Enrollment ON Section.Section_ID = Enrollment.Section_ID
GROUP BY Course.Course_ID, Course.Course_Name
HAVING COUNT(Enrollment.Enrollment_ID) >= 2;

----12. Statement using Conditional Expressions
SELECT Passport_ID,
       First_Name,
```

COURSE_ID	COURSE_NAME	ENROLLMENT_COUNT
5394	BADS	3
4672	DSA	3
5869	FDM	4
4591	JAVA	3
4436	COMARCH	2
5591	INFOVIZ	3

6 rows returned in 0.00 seconds

12.

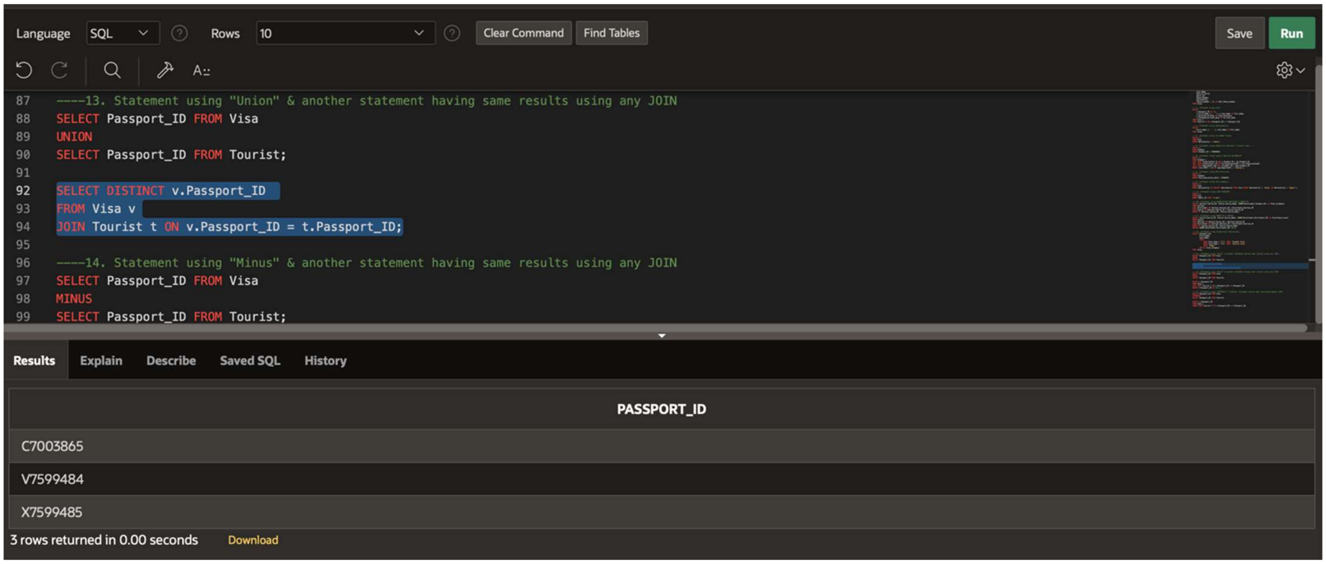


```
----12. Statement using Conditional Expressions
SELECT Passport_ID,
       First_Name,
       Last_Name,
       CASE
           WHEN Visa_Type = 'F-1' THEN 'Student Visa'
           WHEN Visa_Type = 'B-2' THEN 'Tourist Visa'
           ELSE 'Other'
       END AS Visa_Category
FROM Visa;

----13. Statement using "Union" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
```

PASSPORT_ID	FIRST_NAME	LAST_NAME	VISA_CATEGORY
Z3551512	Rohit	Sharma	Student Visa
A7045082	Virat	Kohli	Student Visa
B1820313	Yashaswi	Jaiswal	Student Visa
C7003865	Jasprit	Bumrah	Tourist Visa
U3551512	Kuldeep	Yadav	Student Visa
P7599464	Shivam	Dube	Other
J7599479	Lin	Chen	Student Visa
P7599480	Aisha	Mohamed	Student Visa
L7599481	Thomas	Schmidt	Student Visa
S7599482	Christopher	Lee	Other

More than 10 rows available. Increase rows selector to view more rows.

13.

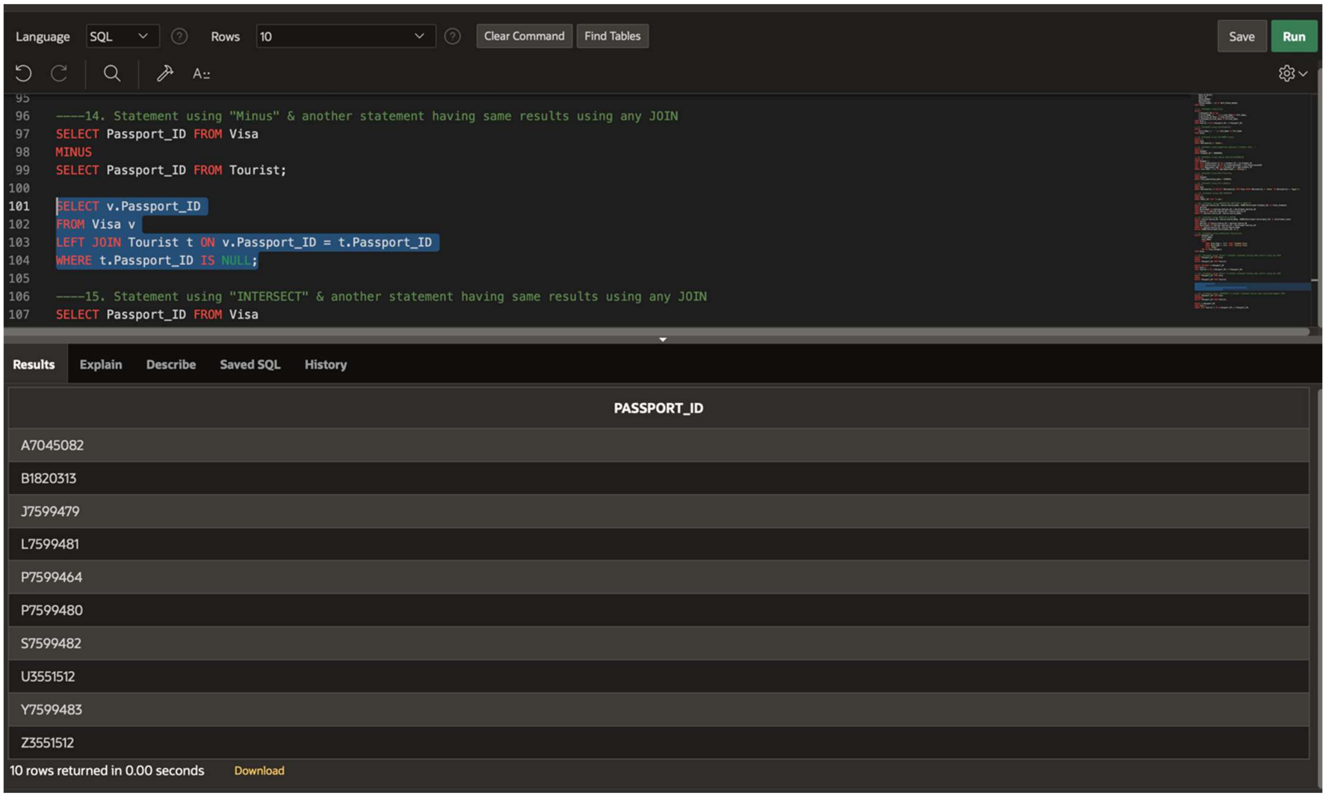


```
----13. Statement using "Union" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
UNION
SELECT Passport_ID FROM Tourist;

SELECT DISTINCT v.Passport_ID
FROM Visa v
JOIN Tourist t ON v.Passport_ID = t.Passport_ID;

----14. Statement using "Minus" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
MINUS
SELECT Passport_ID FROM Tourist;
```

PASSPORT_ID
C7003865
V7599484
X7599485

3 rows returned in 0.00 seconds

14.

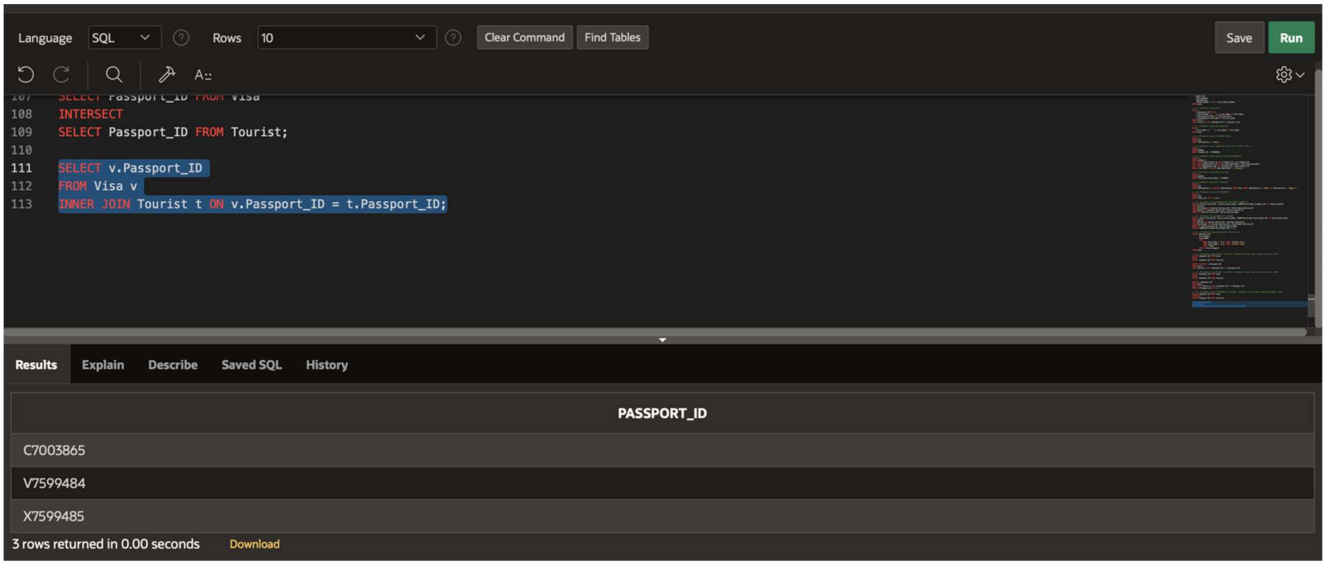


```
----14. Statement using "Minus" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
MINUS
SELECT Passport_ID FROM Tourist;

SELECT v.Passport_ID
FROM Visa v
LEFT JOIN Tourist t ON v.Passport_ID = t.Passport_ID
WHERE t.Passport_ID IS NULL;

----15. Statement using "INTERSECT" & another statement having same results using any JOIN
SELECT Passport_ID FROM Visa
```

PASSPORT_ID
A7045082
B1820313
J7599479
L7599481
P7599464
P7599480
S7599482
U3551512
Y7599483
Z3551512

10 rows returned in 0.00 seconds

15.



```
SELECT Passport_ID FROM Visa
INTERSECT
SELECT Passport_ID FROM Tourist;

SELECT v.Passport_ID
FROM Visa v
INNER JOIN Tourist t ON v.Passport_ID = t.Passport_ID;
```

PASSPORT_ID
C7003865
V7599484
X7599485

3 rows returned in 0.00 seconds Download

PEER TO PEER EVALUATION:

UJJWAL NILAGIRI- 100%

SWARAN ADEPU- 100%

SHALINI KORUPALA- 100%

JASWITHA REDDY DEVIREDDY- 100%

LIKHITHNAGA VENKATA SAI BHAR VYAKARANAM- 100%

SRI BHAVANI MADALA- 100%