DE MONTFORT
UNIVERSITY
LEICESTER

P16218862

JÓZEF KONRAD YIKA

IMAT5103

ASSIGNMENT 1

PRIVATE MUSIC SCHOOL DATABASE

Table of Contents

STAGE 1

TASK 1: SCENARIO

Title: Private Music School ‘Baby Steps’

Private Music School ‘Baby Steps’ is a private music school based in London, UK. The schools’ motto and name encapsulates the belief that individuals who learn how to play the instrument should consider small steps that they make while practicing an important and significant part of the journey.

So far, the school has been operating only in one location (London), but an increasing amount of students and a non-existing database system prompted for necessary changes to be made. The principal decided to open a second branch in Leicester, UK and develop a database for both the locations. There are currently 40 students enrolled and the school wanted to hire new teachers, and other employees to ensure that the student to teacher ratio stays within an acceptable by the school range.

1.a: HOW THE SCHOOL OPERATES?

The school offers one to one music lessons. Those are practice based classes, where a student and a teacher sit in one room playing on the chosen (by a student) instrument. Anyone who is older than 5 can enrol. The duration of one lesson is 60 minutes. How often the student attends the school is determined by the chosen plan. There are two plans offered by the school; full-time and part-time enrolment. Full-time enrolment requires students to attend 5 times a week and part-time 2 times a week. The school operates in two terms; from September to December and from January to June. The tuition fees can be paid as a fixed price for one term (e.g. from September to December), or for an entire year. There is also an option for students to pay per hour. Students don’t have to pay for the missed lessons, if there is a reasonable justification given (such as illness, sickness etc). If the student paid for an entire semester, classes where he/she didn’t attend don’t have to be paid, if there is a reasonable justification given(illness, sickness etc.) The schools’ staff are: head of school (principal), vice-president, administrator, music teachers, cleaners and custodian.

1.b ENROLMENT AND REGISTRATION

Students that are underage (18) must be enrolled by their parents. The enrolment process normally starts with a phone call or in person query. At this stage, the administrator fills out a prospective students’ enrolment form which needs to be signed by the principal who signs the form and admits the student. A student is given a trial-period of 3 lessons where he/she decides whether to stay on the course. A trail-period is paid, however if student decides not to continue, the repayment is made. After the trial-period if the student decided to continue the official registration form is completed with all of the important information such as the students details, payment plan, assigned teacher and enrolment type (full-time, part -time). Students can choose their instrument from all of the instruments families. Every student must take an examination in a form of recital at the end of the term. If the student fails the examination he/she is allowed for up to two retakes. If the attempted retakes also result in failure student is unable to continue the course.

A NEED FOR A DATABASE

As we can see from the points above in order to maintain and keep track of all the records there is need to develop a suitable database that will help to focus more on students and less on the paperwork. This is to ensure that the school may prosper in the future and attract more people.

The most important following data needs to be considered:

- All the employees' details needs to be tracked and updated if necessary. A new employees' data must be registered and current employees' data must be up to date with all of the most important information such as name, last name contact details etc. Music teachers must have assigned students that they teach.
- All students' details needs to be tracked and updated if necessary. A new student must be registered and current students' data must be up to date with all of the most important information such as name, last name, contact details, assigned teacher etc.
- All of the payments made by students need to be tracked and kept up to date.
- All the data from the 3 points above needs to be replicated for the second branch in Leicester, UK.

TASK 2: CONCEPTUAL DESIGN

ENTITIES AND ATTRIBUTES

The following presents **entities** (**tables**) and their **attributes** (**columns**) as a starting point for the conceptual design (entities on the left, attributes on the right in the square brackets).

Note: This is pre *'Normalization'* and may slightly change while creating a logical design.

1.**Students** [First Name, Last Name, Gender, Street Address , City, Postcode, Phone number, Email, Assigned Teacher, Assigned Instrument, Chosen Study Type, Chosen Tuition Fee Plan]

2.**Music Teachers** [First Name, Last Name, Gender, Street Address , City, Postcode, Phone number, Email, Assigned Students, Specialty (Instrument), Employee Category]

3.**Employee** [First Name, Last Name, Phone Number, Street Address, City, Postcode, Email]

4.**Branch** [Location, Postcode, Street Address, City, Contact Details [Phone Number, Email]]

5.**TuitionFeePayment**[This table will contain information about the payments made by students such as payment date, payment number, type of the payment [credit card, in person]]

6.**Enrolment** [contains details about the students who enrolled to the school]

7.**Registration**[contains details about the students who officially registered after the trial period]

8.**Instruments**[contains details about all of the instruments available at the school]

9.**Outstanding Payments**[contains details about all of the outstanding students' payments]

10.**Attendance**[contains details about the students' attendance]

11. **Absence From Class** [contains details about the number of times a student didn't attend the class during the academic year]

12. **Enrolment Declined**[contains details about the number of students who decided not to continue after the trial period]

13. **Examinations**[contains details about all of the students that take examinations at the end of the term]

14. **Principal[** contains details about principal and the number of students the principal admitted]

15. **Administrator**[**contains** details about the administrator and the enrolment form]

16. **Form [** contains details about the school forms such as registration, enrolment]

ASSUMPTIONS

1.Branch in London is managed by a principal and in Leicester by Vice-Principal. Principal and vice-principal are both considered staff members/employees. This means that one and only one employee can manage/supervise exactly one branch, but not all employees manage the branch.

2.Employees can be a music teacher, a principal, an administrator or anyone mentioned above in the scenario.

3.One student can be assigned to only one teacher, however one teacher can have assigned at least 1 and up to 5 students.

4.One student can take up to three examinations per year (one at the end of the first term + 2 retakes , one at the end of the second term + 2 retakes)

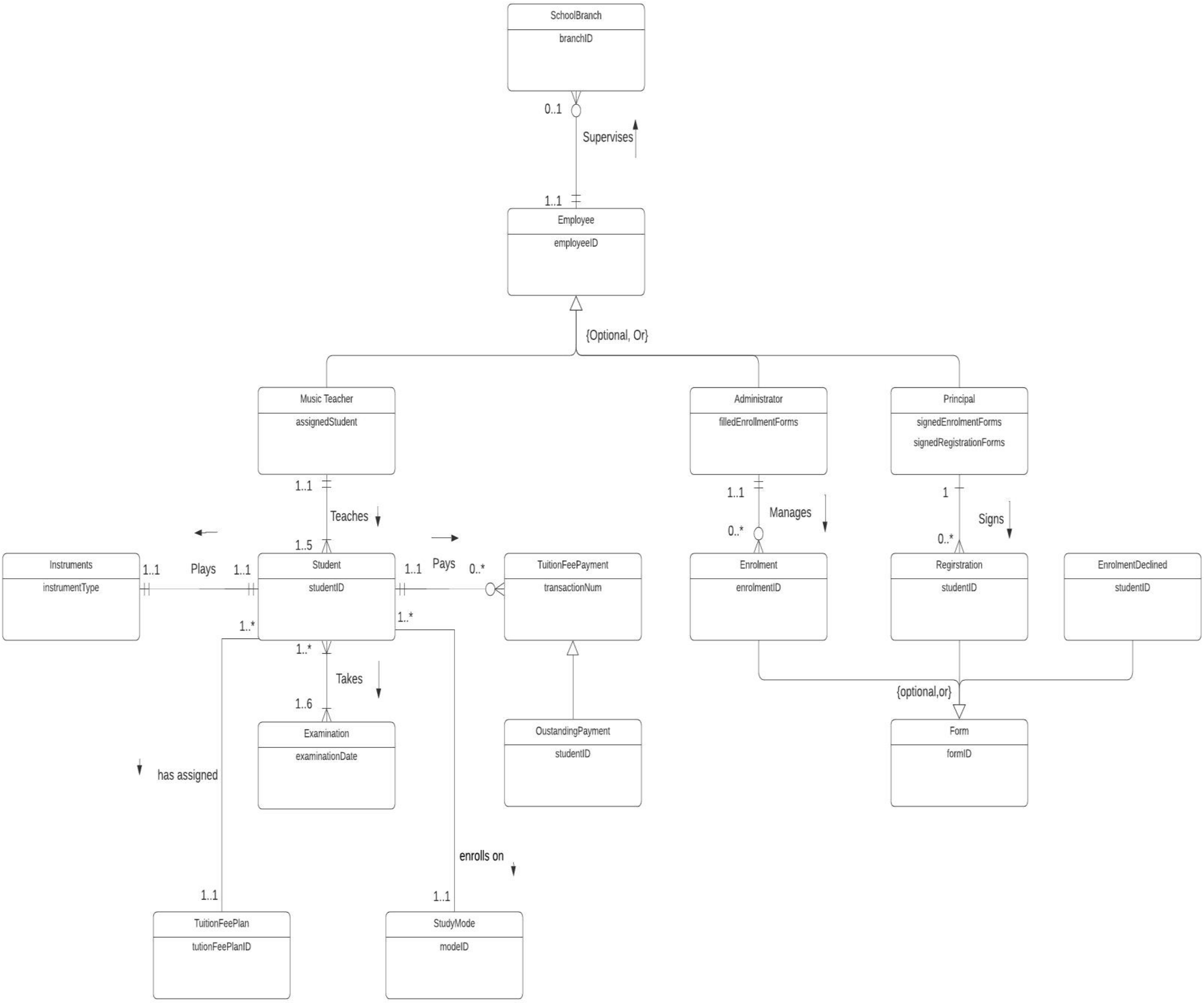
5.One Administrator can be in charge of 0 or many enrolments.

6.A principal can sign 0 or more registration forms.

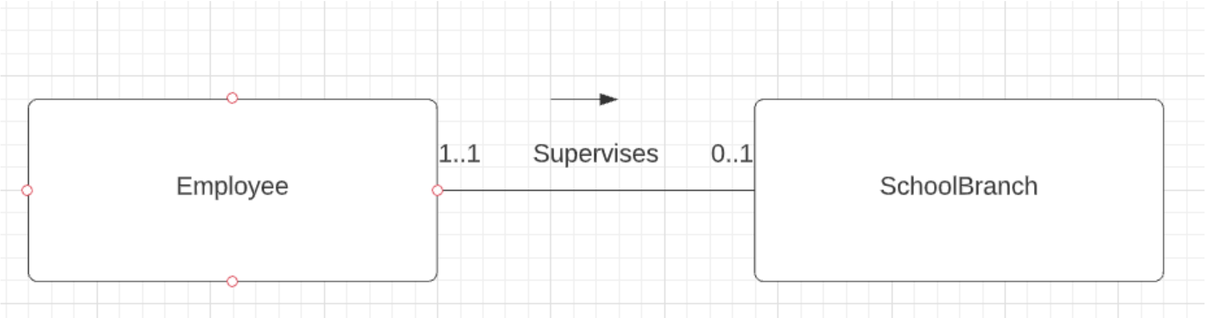
7. A student can play only on one instrument. There can’t be assigned many instruments to one student.

8. One student can have zero or many payments.

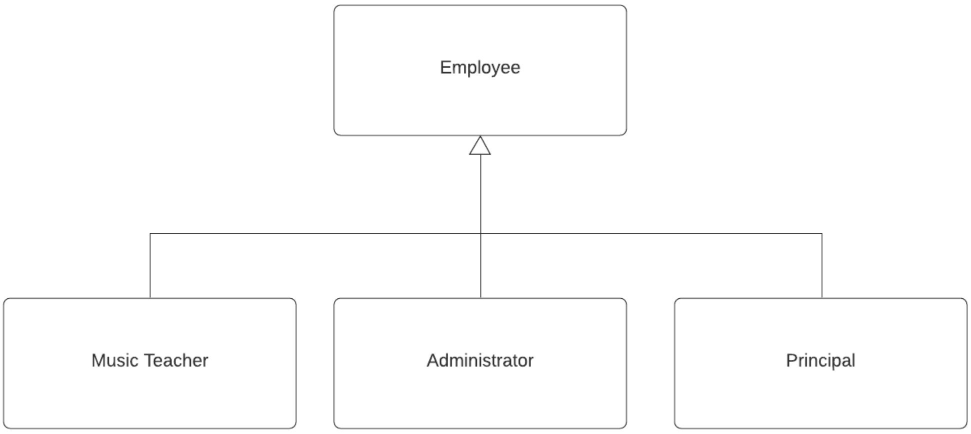
EER DIAGRAM



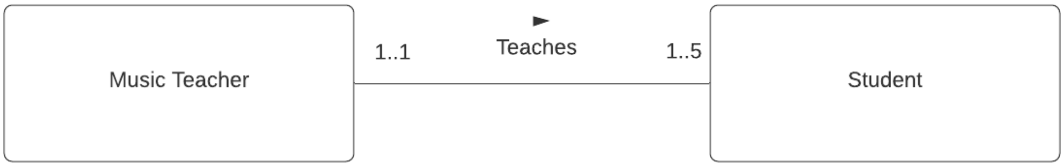
ENTERPIRSE RULES



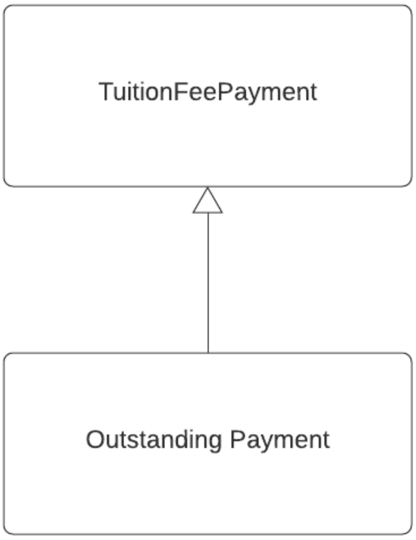
Only 1 employee can supervise one branch. More than 1 employee can't supervise 1 branch.



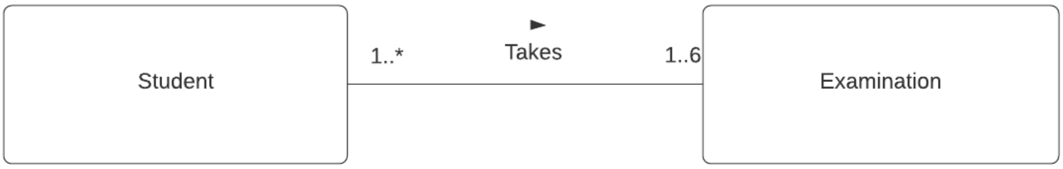
Music Teacher, Adminstrator and Principal are type of employees within the school.



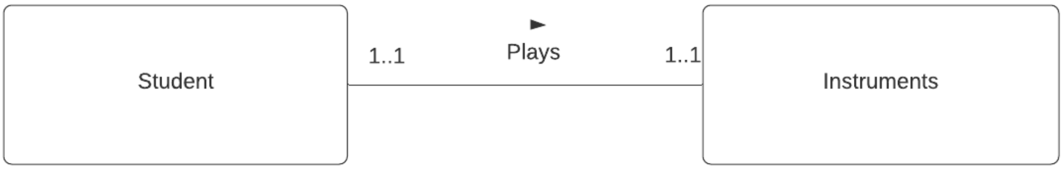
One Music Teacher can teach at least 1 and up to 5 student.



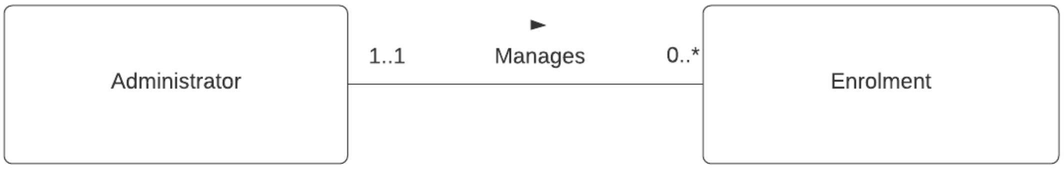
An Outstanding payment is a type of payment.



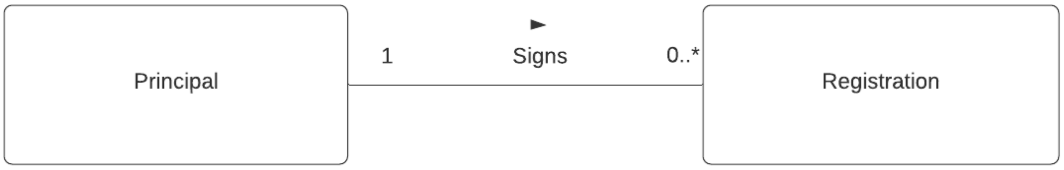
One ore more students can take at least 1 and up to 6 examinations during the year.



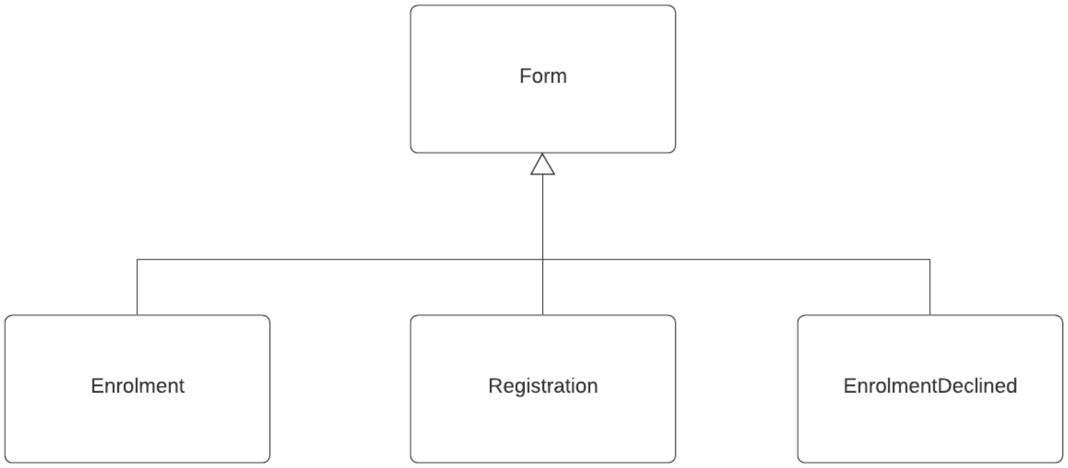
One student can only play on one instruments.



One Administrator can manage 0 or more student enrolments



1 Principal signs 0 or many registration forms



Enrolment,Registration and EnrolmentDeclined are all types of Form.

STAGE 2

TASK 3: LOGICAL DATABASE DESIGN

LOGICAL DATABASE DESIGN

The following presents a logical normalized database design for the Private Music School Scenario with all of the entities and attributes.

Note: Entities on the left – Attributes on the right – Primary Keys in bold and underlined – Foreign Keys in Italics – Non Key attributes in Calibri

1.Student [**studentID,** firstName, lastName, gender, streetAddress, city, postcode, phoneNumber, email, *teacherID , studyMode, tuitionFeePlan*]

2.Music Teacher [**teacherID**, firstName, lastName, gender, streetAddress, city, postcode, phoneNumber, email]

3. Instrument [**instrumentID,** instrumentType, instrumentFamily, *studentID*]

4.StudyMode [**modeID**, modeType [fulltime/partime]]

5.Employee [**employeeID**, firstName, lastName, phoneNumber, street, city, postcode, email, *branchID*]

6.Branch [**branchID**, postcode, street, city, phoneNumber, email]

7.TuitionFeePlan[**tuitionFeePlanNumber**, tuitionFeeType]

8.TuitionFeePayment[**tuitionFeePaymentReferenceNumber**, *tuitionfeeType*, *studentID*, paymentDate]

9.Outstanding Payment**[paymentReferenceNumber**, studentID, amount]

10. Examinations**[** examTime, examDate***, studentID***]

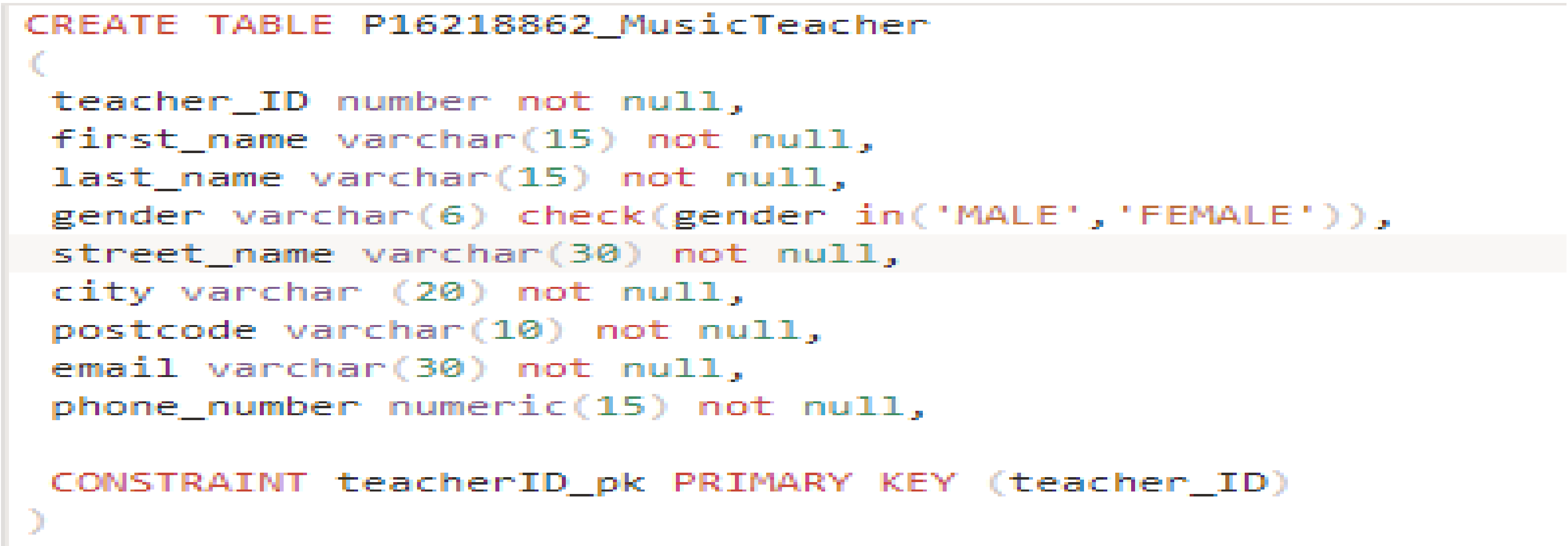
11. Principal**[principalName**, lastName, phoneNumber, email, *branchID*, numOfSignedForm]

12. Administrator[**administratorID**, firstName, lastName, *branchID*]

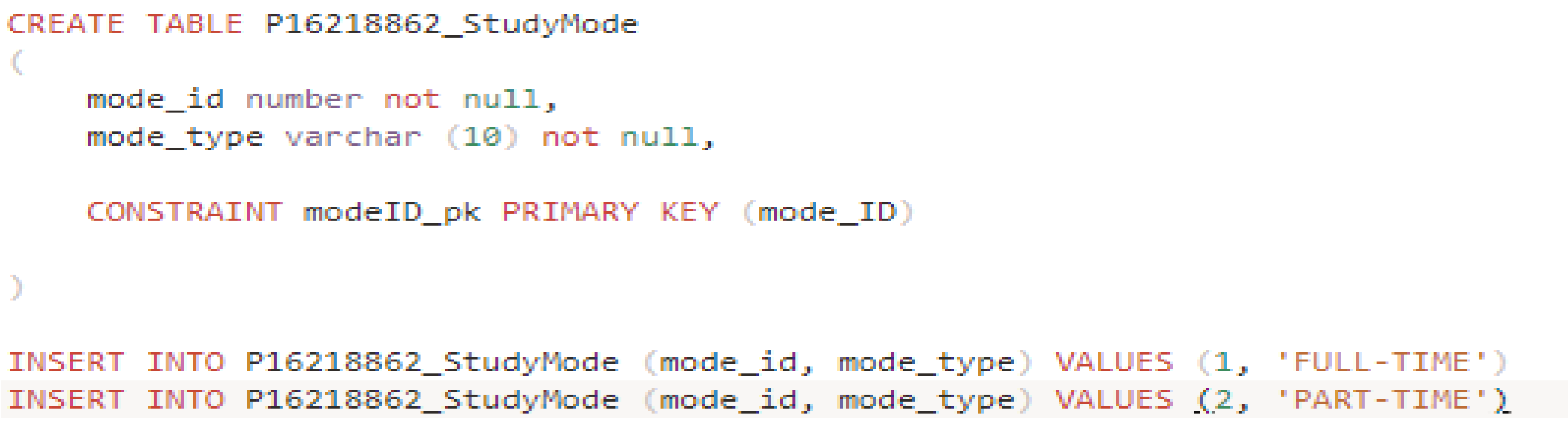
13. Form **[formID,** administratorID, formType]

TASK 4: ORACLE DBMS TABLES

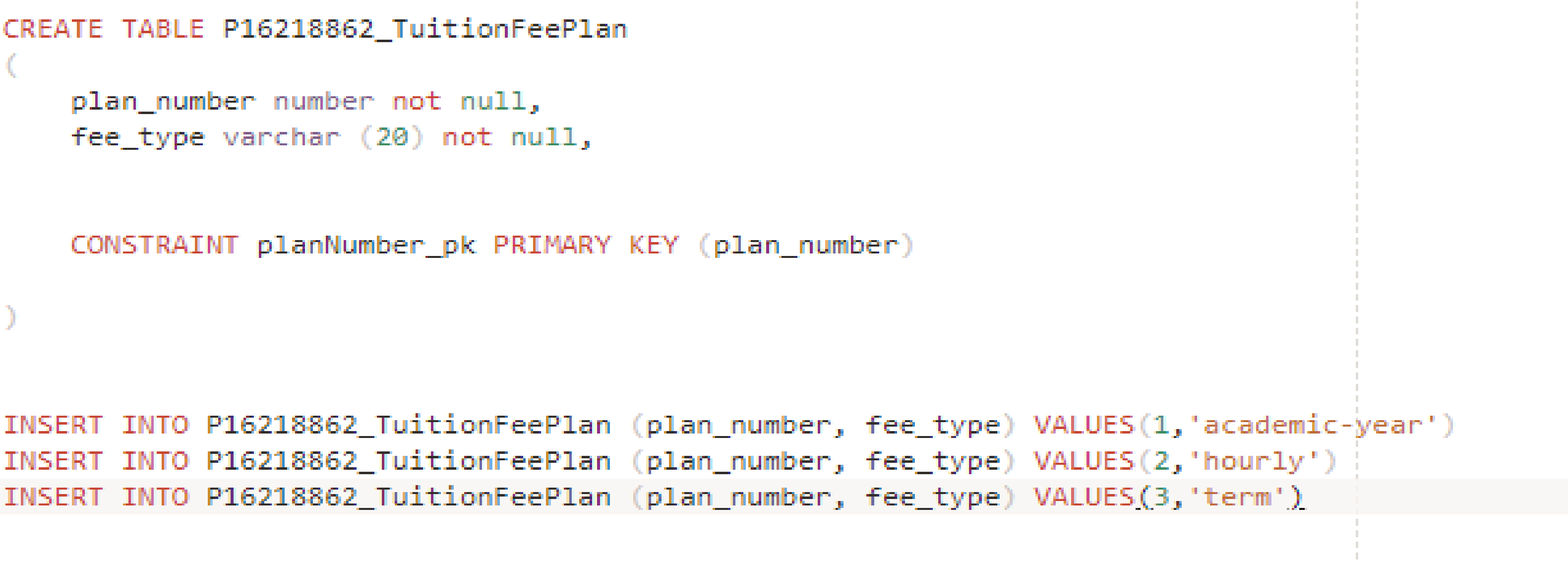
The following screenshots present the tables and their sample data.



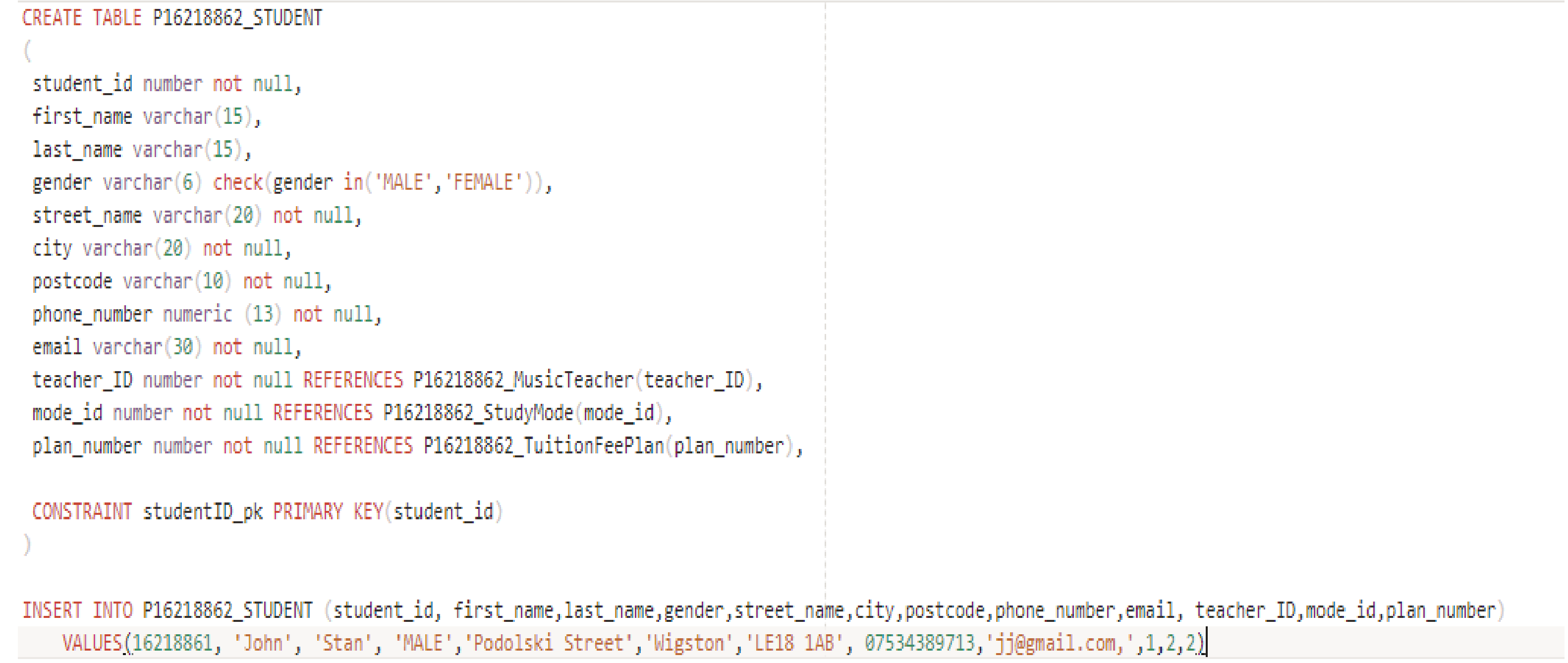
```
CREATE TABLE P16218862_MusicTeacher
(
 teacher_ID number not null,
 first_name varchar(15) not null,
 last_name varchar(15) not null,
 gender varchar(6) check(gender in('MALE','FEMALE')),
 street_name varchar(30) not null,
 city varchar (20) not null,
 postcode varchar(10) not null,
 email varchar(30) not null,
 phone_number numeric(15) not null,

 CONSTRAINT teacherID_pk PRIMARY KEY (teacher_ID)
)
```



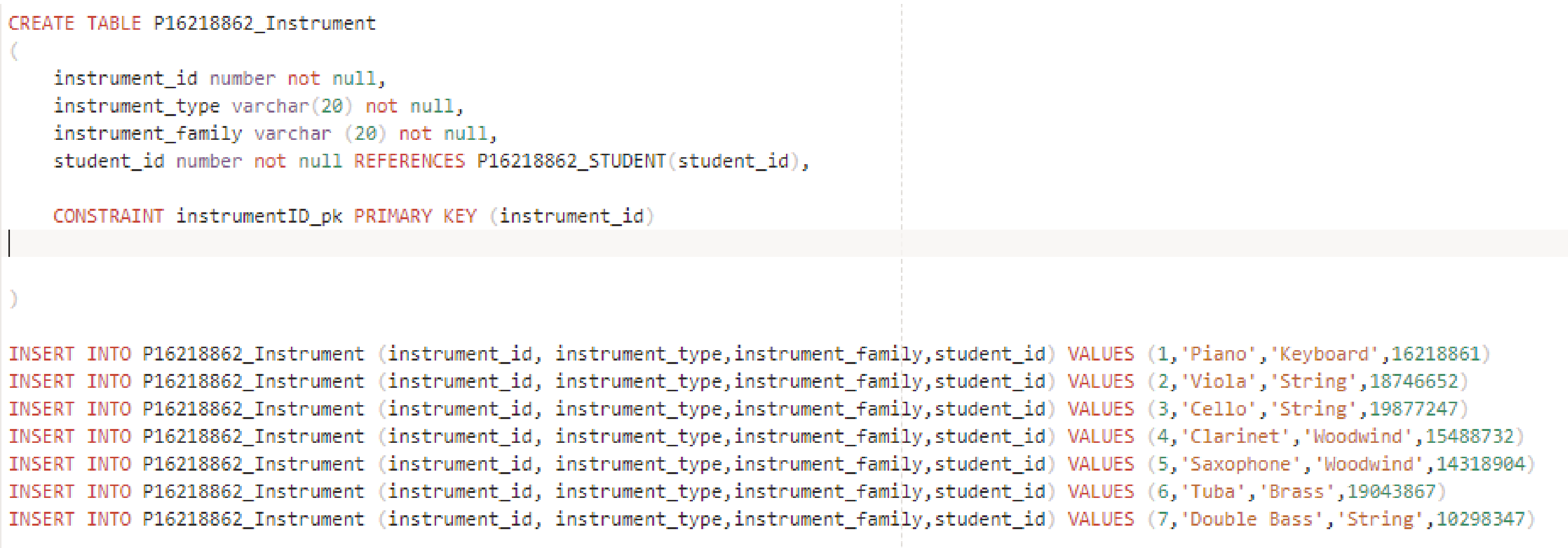
```
CREATE TABLE P16218862_StudyMode
(
     mode_id number not null,
     mode_type varchar (10) not null,

     CONSTRAINT modeID_pk PRIMARY KEY (mode_ID)

)

INSERT INTO P16218862_StudyMode (mode_id, mode_type) VALUES (1, 'FULL-TIME')
INSERT INTO P16218862_StudyMode (mode_id, mode_type) VALUES (2, 'PART-TIME')
```



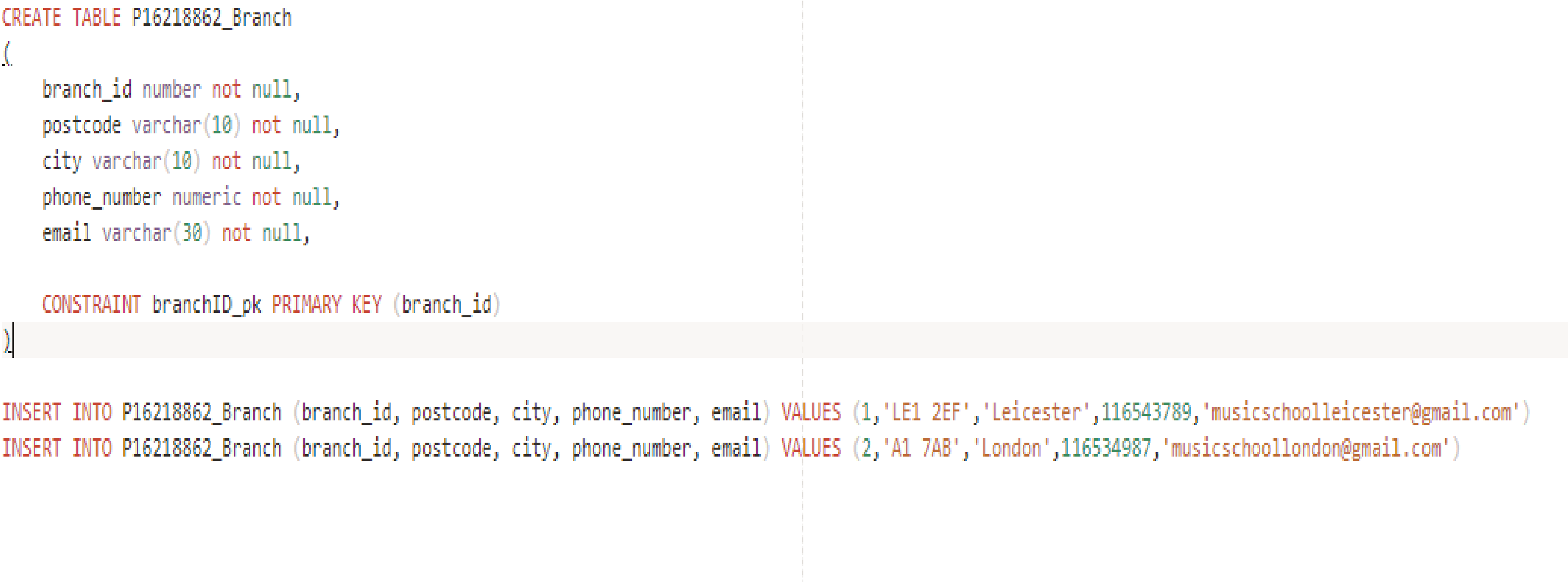
```
CREATE TABLE P16218862_TuitionFeePlan
(
    plan_number number not null,
    fee_type varchar (20) not null,


    CONSTRAINT planNumber_pk PRIMARY KEY (plan_number)

)

INSERT INTO P16218862_TuitionFeePlan (plan_number, fee_type) VALUES(1,'academic-year')
INSERT INTO P16218862_TuitionFeePlan (plan_number, fee_type) VALUES(2,'hourly')
INSERT INTO P16218862_TuitionFeePlan (plan_number, fee_type) VALUES(3,'term')
```



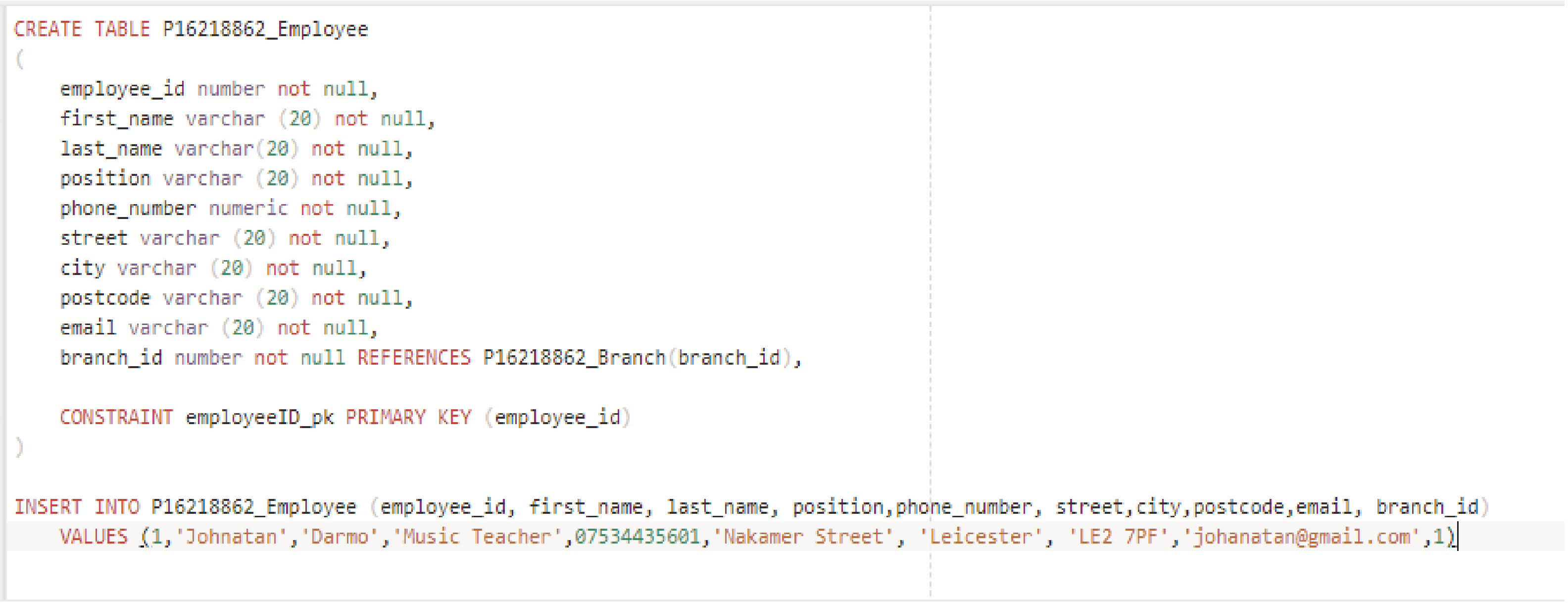
```
CREATE TABLE P16218862_STUDENT
(
 student_id number not null,
 first_name varchar(15),
 last_name varchar(15),
 gender varchar(6) check(gender in('MALE','FEMALE')),
 street_name varchar(20) not null,
 city varchar(20) not null,
 postcode varchar(10) not null,
 phone_number numeric (13) not null,
 email varchar(30) not null,
 teacher_ID number not null REFERENCES P16218862_MusicTeacher(teacher_ID),
 mode_id number not null REFERENCES P16218862_StudyMode(mode_id),
 plan_number number not null REFERENCES P16218862_TuitionFeePlan(plan_number),

 CONSTRAINT studentID_pk PRIMARY KEY(student_id)
)

INSERT INTO P16218862_STUDENT (student_id, first_name,last_name,gender,street_name,city,postcode,phone_number,email, teacher_ID,mode_id,plan_number)
    VALUES(16218861, 'John', 'Stan', 'MALE','Podolski Street','Wigston','LE18 1AB', 07534389713,'jj@gmail.com,',1,2,2)
```



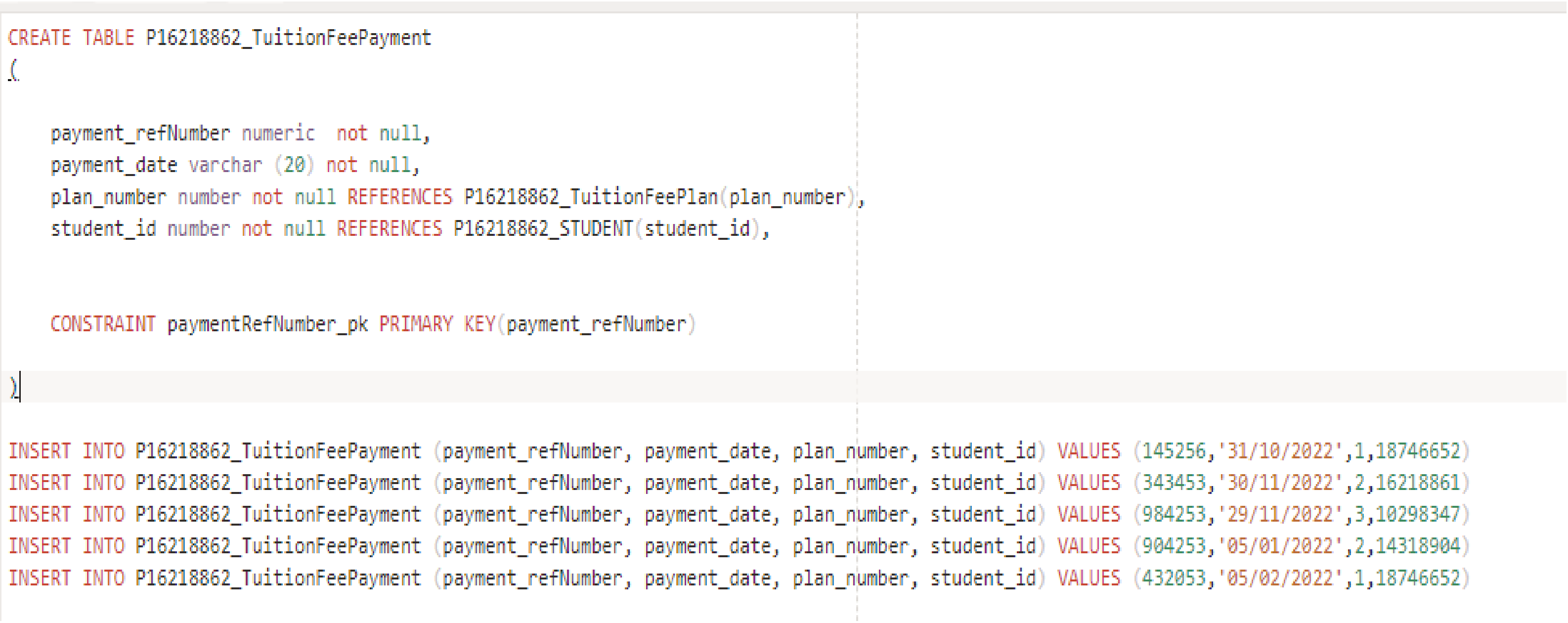
```
CREATE TABLE P16218862_Instrument
(
    instrument_id number not null,
    instrument_type varchar(20) not null,
    instrument_family varchar (20) not null,
    student_id number not null REFERENCES P16218862_STUDENT(student_id),

    CONSTRAINT instrumentID_pk PRIMARY KEY (instrument_id)


)

INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (1,'Piano','Keyboard',16218861)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (2,'Viola','String',18746652)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (3,'Cello','String',19877247)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (4,'Clarinet','Woodwind',15488732)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (5,'Saxophone','Woodwind',14318904)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (6,'Tuba','Brass',19043867)
INSERT INTO P16218862_Instrument (instrument_id, instrument_type,instrument_family,student_id) VALUES (7,'Double Bass','String',10298347)
```



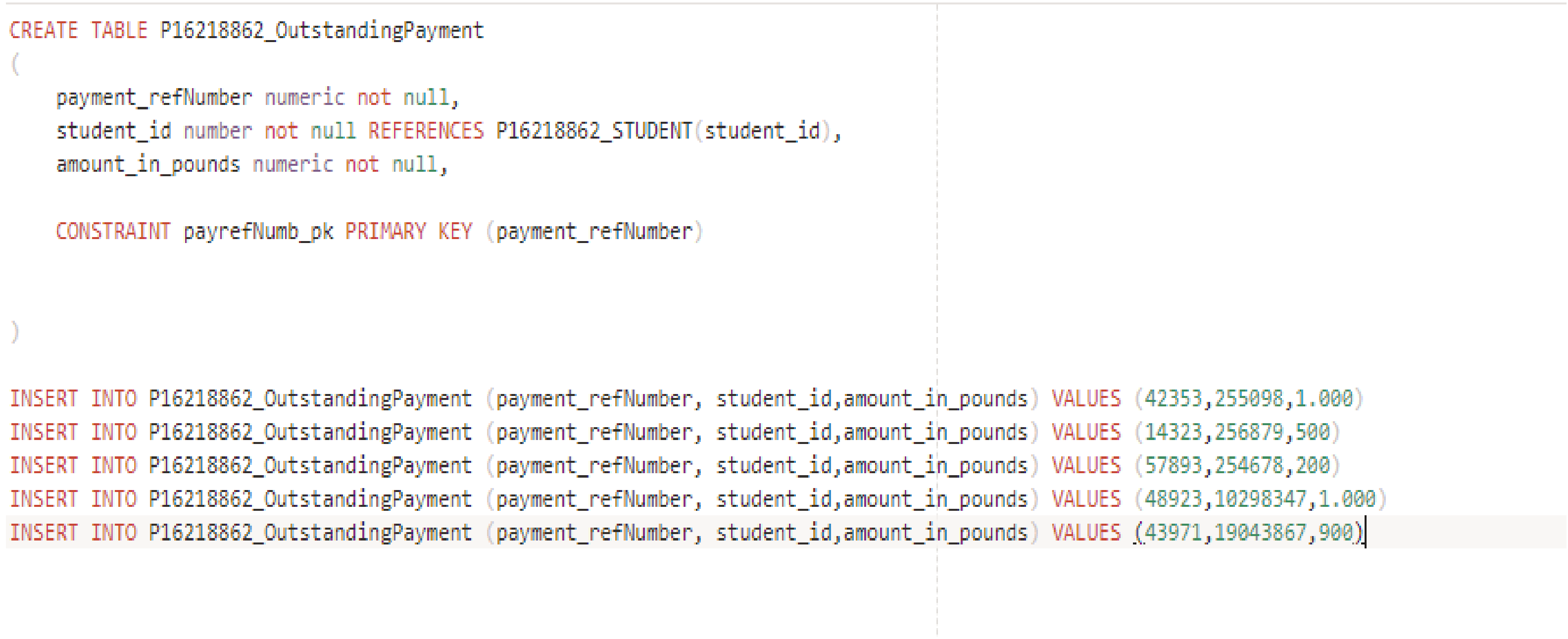
```
CREATE TABLE P16218862_Branch
(
    branch_id number not null,
    postcode varchar(10) not null,
    city varchar(10) not null,
    phone_number numeric not null,
    email varchar(30) not null,

    CONSTRAINT branchID_pk PRIMARY KEY (branch_id)
)

INSERT INTO P16218862_Branch (branch_id, postcode, city, phone_number, email) VALUES (1,'LE1 2EF','Leicester',116543789,'musicschoolleicester@gmail.com')
INSERT INTO P16218862_Branch (branch_id, postcode, city, phone_number, email) VALUES (2,'A1 7AB','London',116534987,'musicschoollondon@gmail.com')
```



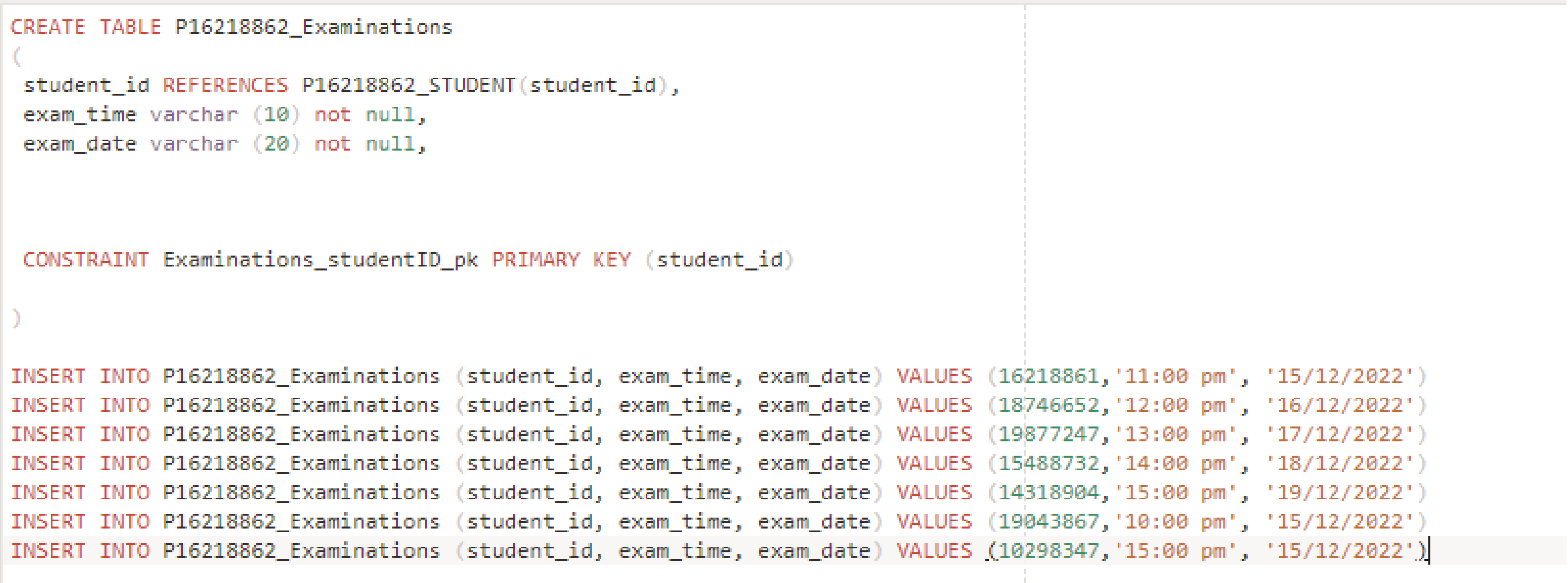
```sql
CREATE TABLE P16218862_Employee
(
    employee_id number not null,
    first_name varchar (20) not null,
    last_name varchar(20) not null,
    position varchar (20) not null,
    phone_number numeric not null,
    street varchar (20) not null,
    city varchar (20) not null,
    postcode varchar (20) not null,
    email varchar (20) not null,
    branch_id number not null REFERENCES P16218862_Branch(branch_id),

    CONSTRAINT employeeID_pk PRIMARY KEY (employee_id)
)

INSERT INTO P16218862_Employee (employee_id, first_name, last_name, position,phone_number, street,city,postcode,email, branch_id)
    VALUES (1,'Johnatan','Darmo','Music Teacher',07534435601,'Nakamer Street', 'Leicester', 'LE2 7PF','johanatan@gmail.com',1)
```



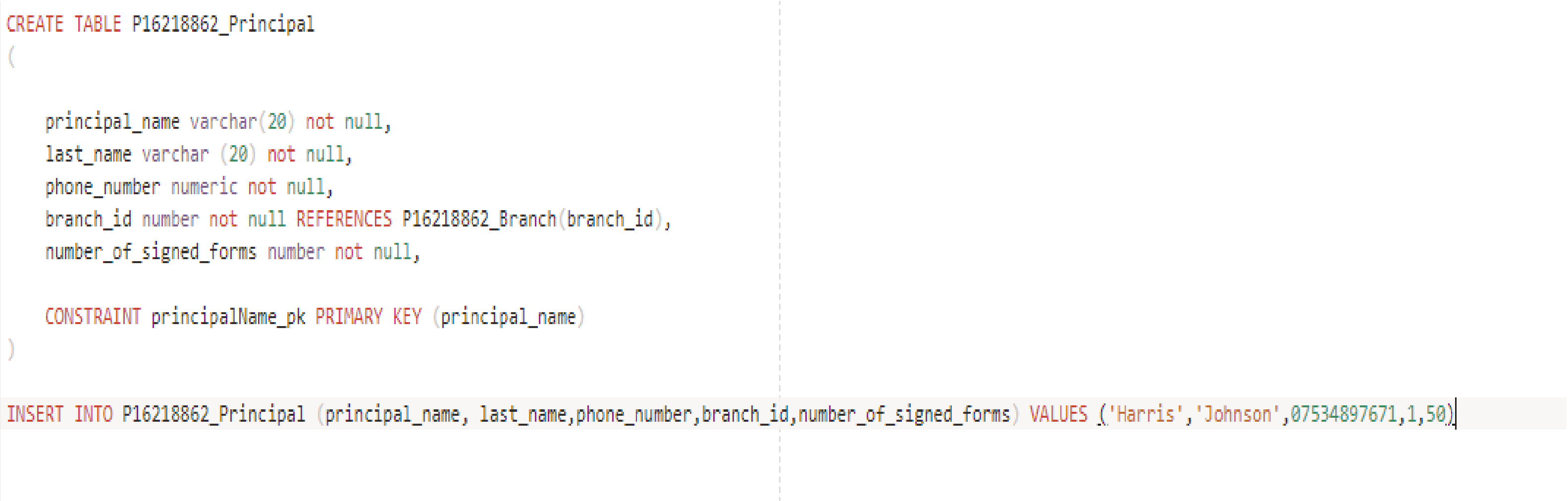
```sql
CREATE TABLE P16218862_TuitionFeePayment
(

    payment_refNumber numeric  not null,
    payment_date varchar (20) not null,
    plan_number number not null REFERENCES P16218862_TuitionFeePlan(plan_number),
    student_id number not null REFERENCES P16218862_STUDENT(student_id),


    CONSTRAINT paymentRefNumber_pk PRIMARY KEY(payment_refNumber)

)

INSERT INTO P16218862_TuitionFeePayment (payment_refNumber, payment_date, plan_number, student_id) VALUES (145256,'31/10/2022',1,18746652)
INSERT INTO P16218862_TuitionFeePayment (payment_refNumber, payment_date, plan_number, student_id) VALUES (343453,'30/11/2022',2,16218861)
INSERT INTO P16218862_TuitionFeePayment (payment_refNumber, payment_date, plan_number, student_id) VALUES (984253,'29/11/2022',3,10298347)
INSERT INTO P16218862_TuitionFeePayment (payment_refNumber, payment_date, plan_number, student_id) VALUES (904253,'05/01/2022',2,14318904)
INSERT INTO P16218862_TuitionFeePayment (payment_refNumber, payment_date, plan_number, student_id) VALUES (432053,'05/02/2022',1,18746652)
```



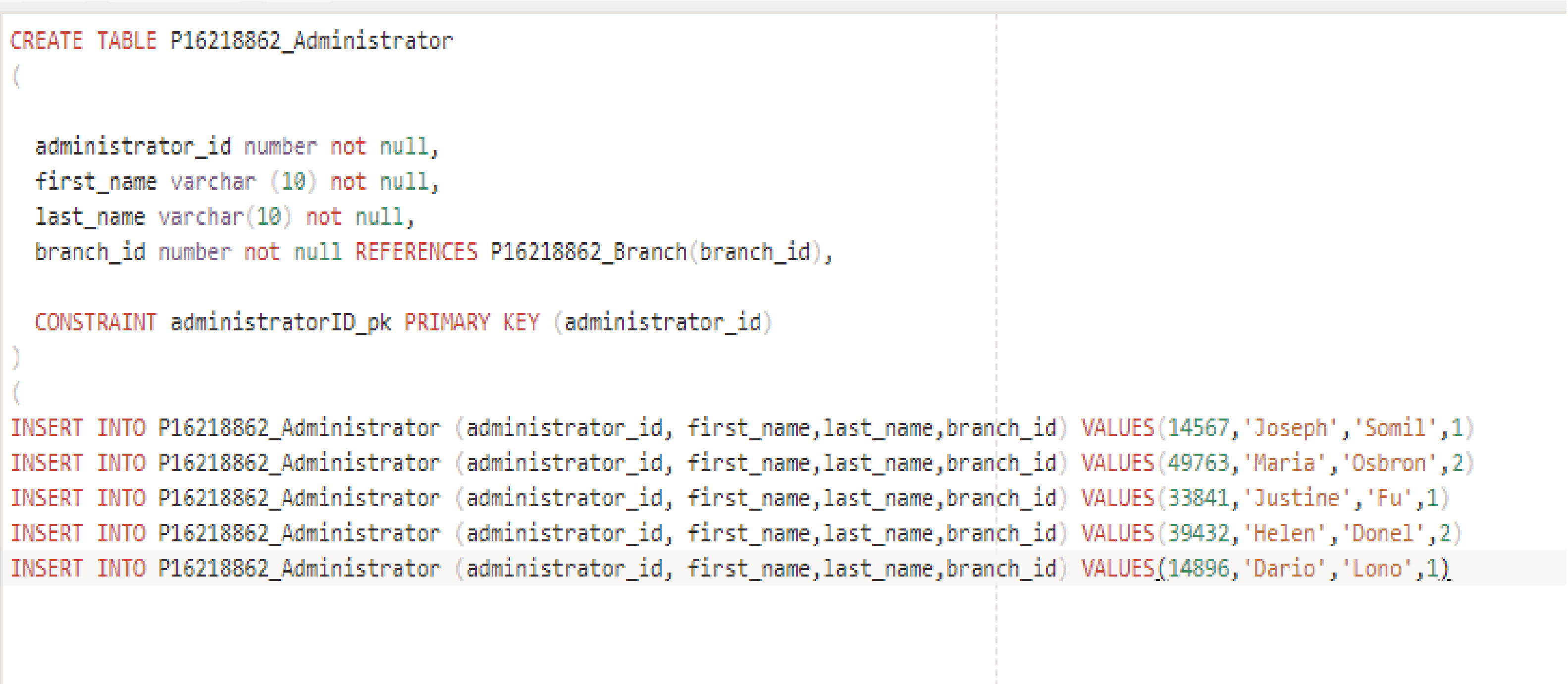
```sql
CREATE TABLE P16218862_OutstandingPayment
(
    payment_refNumber numeric not null,
    student_id number not null REFERENCES P16218862_STUDENT(student_id),
    amount_in_pounds numeric not null,

    CONSTRAINT payrefNumb_pk PRIMARY KEY (payment_refNumber)


)

INSERT INTO P16218862_OutstandingPayment (payment_refNumber, student_id,amount_in_pounds) VALUES (42353,255098,1.000)
INSERT INTO P16218862_OutstandingPayment (payment_refNumber, student_id,amount_in_pounds) VALUES (14323,256879,500)
INSERT INTO P16218862_OutstandingPayment (payment_refNumber, student_id,amount_in_pounds) VALUES (57893,254678,200)
INSERT INTO P16218862_OutstandingPayment (payment_refNumber, student_id,amount_in_pounds) VALUES (48923,10298347,1.000)
INSERT INTO P16218862_OutstandingPayment (payment_refNumber, student_id,amount_in_pounds) VALUES (43971,19043867,900)
```



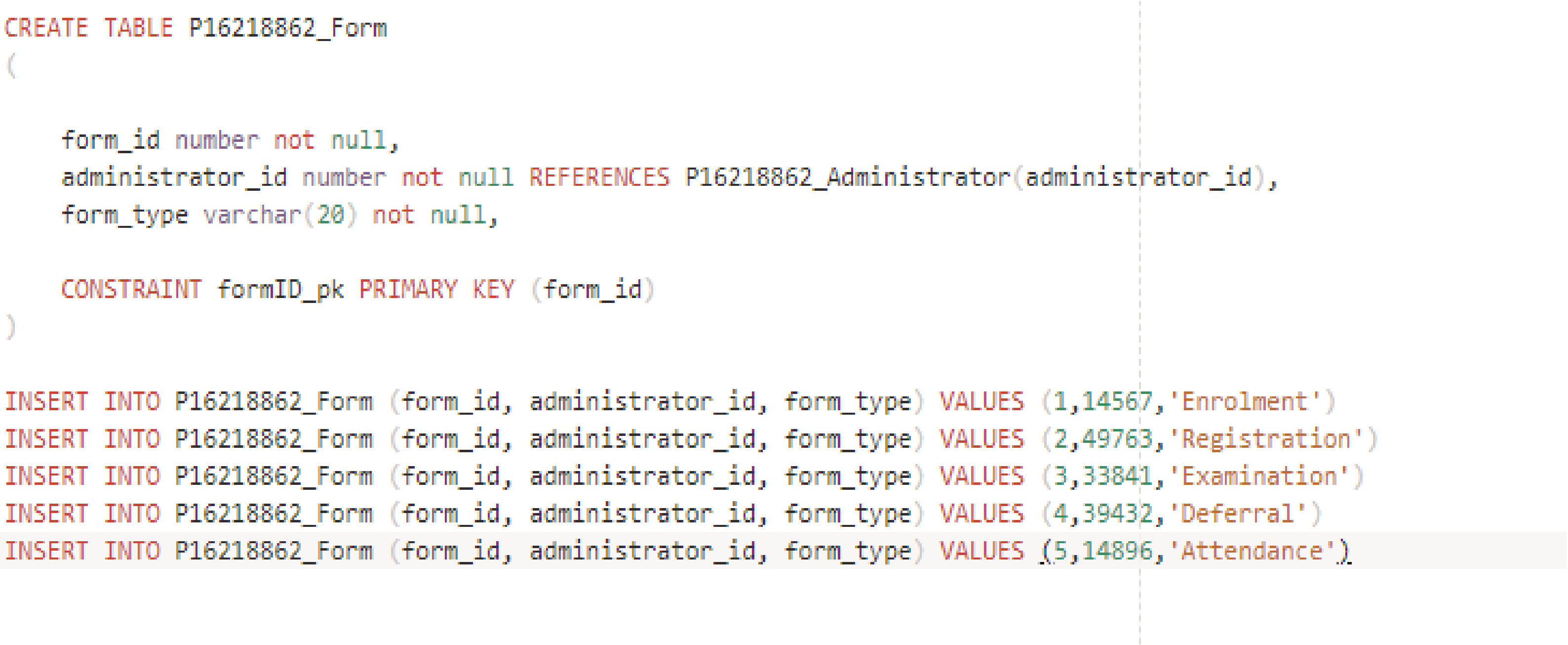
```
CREATE TABLE P16218862_Examinations
(
 student_id REFERENCES P16218862_STUDENT(student_id),
 exam_time varchar (10) not null,
 exam_date varchar (20) not null,



 CONSTRAINT Examinations_studentID_pk PRIMARY KEY (student_id)

)

INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (16218861,'11:00 pm', '15/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (18746652,'12:00 pm', '16/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (19877247,'13:00 pm', '17/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (15488732,'14:00 pm', '18/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (14318904,'15:00 pm', '19/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (19043867,'10:00 pm', '15/12/2022')
INSERT INTO P16218862_Examinations (student_id, exam_time, exam_date) VALUES (10298347,'15:00 pm', '15/12/2022')
```



```
CREATE TABLE P16218862_Principal
(

    principal_name varchar(20) not null,
    last_name varchar (20) not null,
    phone_number numeric not null,
    branch_id number not null REFERENCES P16218862_Branch(branch_id),
    number_of_signed_forms number not null,

    CONSTRAINT principalName_pk PRIMARY KEY (principal_name)
)

INSERT INTO P16218862_Principal (principal_name, last_name,phone_number,branch_id,number_of_signed_forms) VALUES ('Harris','Johnson',07534897671,1,50)
```



```
CREATE TABLE P16218862_Administrator
(

  administrator_id number not null,
  first_name varchar (10) not null,
  last_name varchar(10) not null,
  branch_id number not null REFERENCES P16218862_Branch(branch_id),

  CONSTRAINT administratorID_pk PRIMARY KEY (administrator_id)
)
(
INSERT INTO P16218862_Administrator (administrator_id, first_name,last_name,branch_id) VALUES(14567,'Joseph','Somil',1)
INSERT INTO P16218862_Administrator (administrator_id, first_name,last_name,branch_id) VALUES(49763,'Maria','Osbron',2)
INSERT INTO P16218862_Administrator (administrator_id, first_name,last_name,branch_id) VALUES(33841,'Justine','Fu',1)
INSERT INTO P16218862_Administrator (administrator_id, first_name,last_name,branch_id) VALUES(39432,'Helen','Donel',2)
INSERT INTO P16218862_Administrator (administrator_id, first_name,last_name,branch_id) VALUES(14896,'Dario','Lono',1)
```



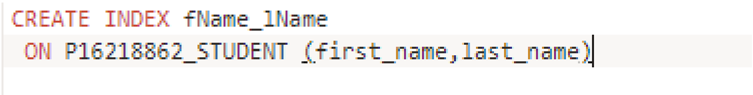
```
CREATE TABLE P16218862_Form
(

    form_id number not null,
    administrator_id number not null REFERENCES P16218862_Administrator(administrator_id),
    form_type varchar(20) not null,

    CONSTRAINT formID_pk PRIMARY KEY (form_id)
)

INSERT INTO P16218862_Form (form_id, administrator_id, form_type) VALUES (1,14567,'Enrolment')
INSERT INTO P16218862_Form (form_id, administrator_id, form_type) VALUES (2,49763,'Registration')
INSERT INTO P16218862_Form (form_id, administrator_id, form_type) VALUES (3,33841,'Examination')
INSERT INTO P16218862_Form (form_id, administrator_id, form_type) VALUES (4,39432,'Deferral')
INSERT INTO P16218862_Form (form_id, administrator_id, form_type) VALUES (5,14896,'Attendance')
```

TASK 5: INDEXES

Indexes help us retrieve specific data faster. They are based on the same principle as an index in the book. When we try to retrieve an information about a certain person in the book, the fastest way to find the corresponding section is to look at the indexes page with the person’s name. Indexes in SQL work in a similar manner.

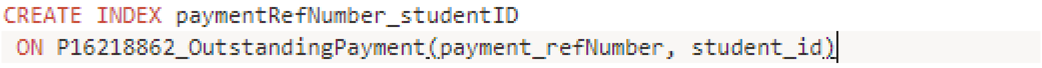
In an institutions such as school or university, the data that will be often retrieved and updates concerns students. Therefore, most of the indexes for this project were created with this in mind.

1. First Name and Last name of a student are important information that will be retrieved quite often.



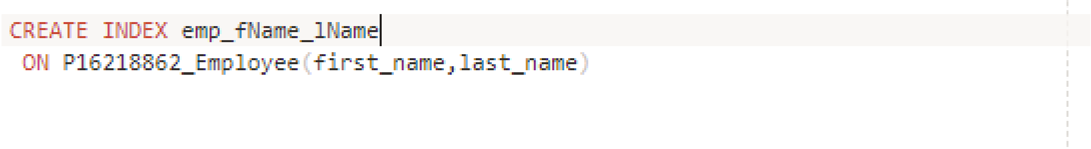
```
CREATE INDEX fName_lName
 ON P16218862_STUDENT (first_name,last_name)
```

2. Payments and its’ associated student is another example of data that needs to be retrieved on a regular basis.



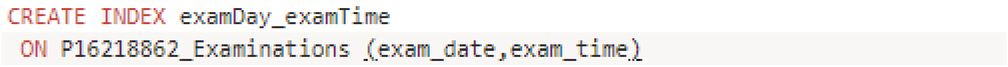
```
CREATE INDEX paymentRefNumber_studentID
 ON P16218862_OutstandingPayment(payment_refNumber, student_id)
```

3. It’s also important to know employees’ names.



```
CREATE INDEX emp_fName_lName
 ON P16218862_Employee(first_name,last_name)
```

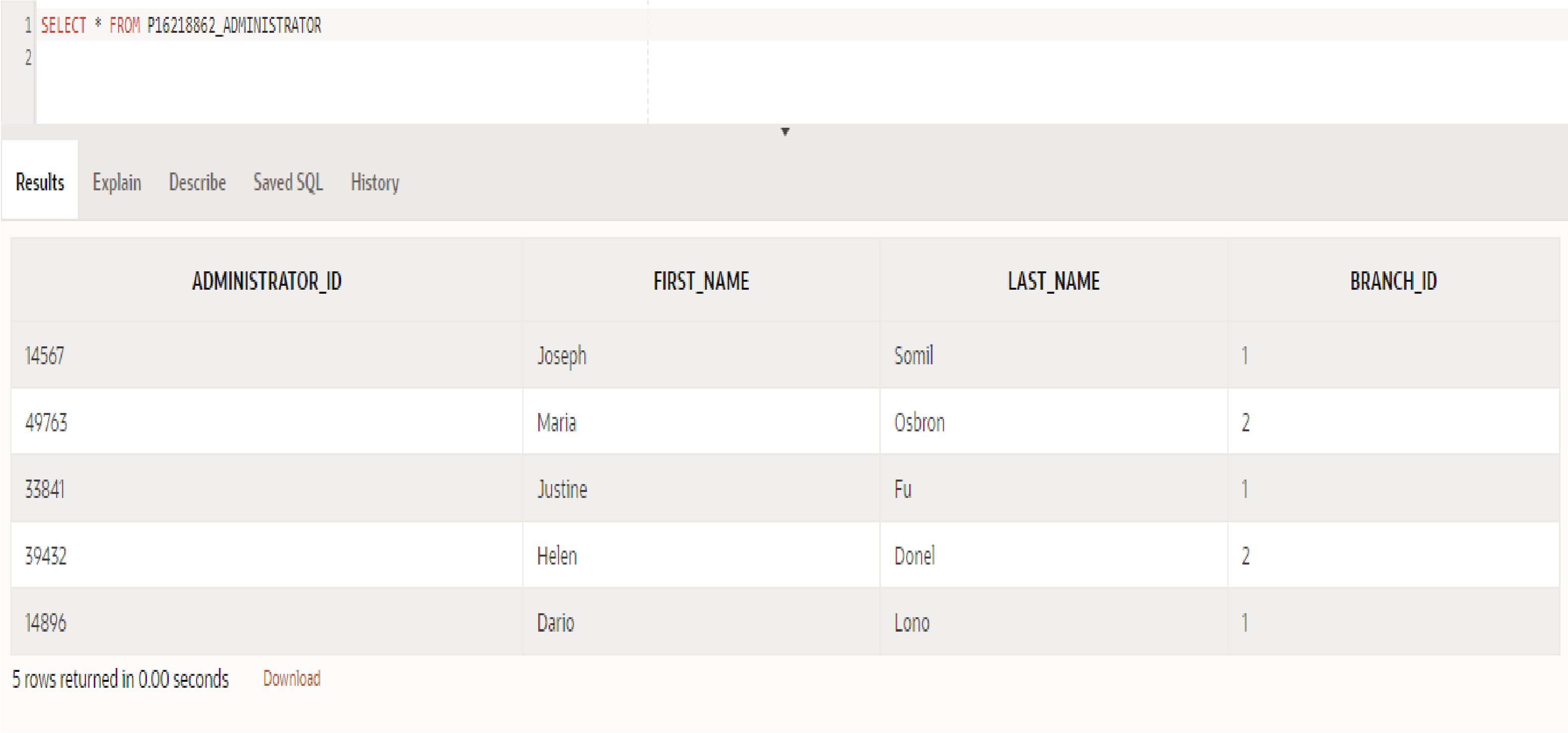
4. The day and time of the exam will be also often retrieved.



```
CREATE INDEX examDay_examTime
 ON P16218862_Examinations (exam_date,exam_time)
```

TASK 6: DATA POPULATION

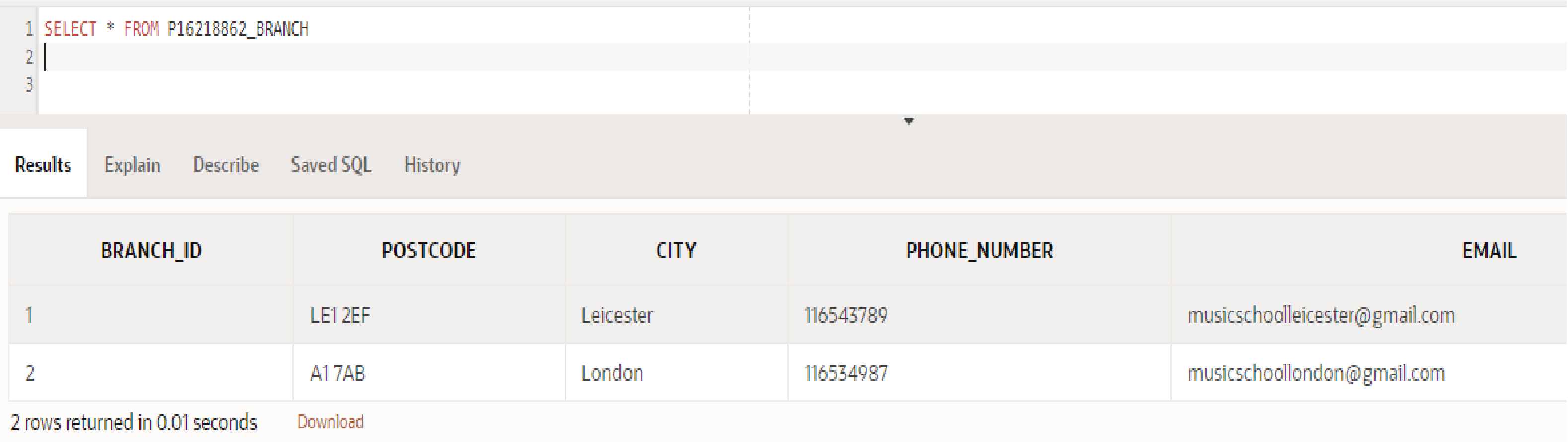
The following pictures present the data that was inserted into the tables.



```
SELECT * FROM P16218862_ADMINISTRATOR
```

ADMINISTRATOR_ID	FIRST_NAME	LAST_NAME	BRANCH_ID
14567	Joseph	Somil	1
49763	Maria	Osbron	2
33841	Justine	Fu	1
39432	Helen	Donel	2
14896	Dario	Lono	1

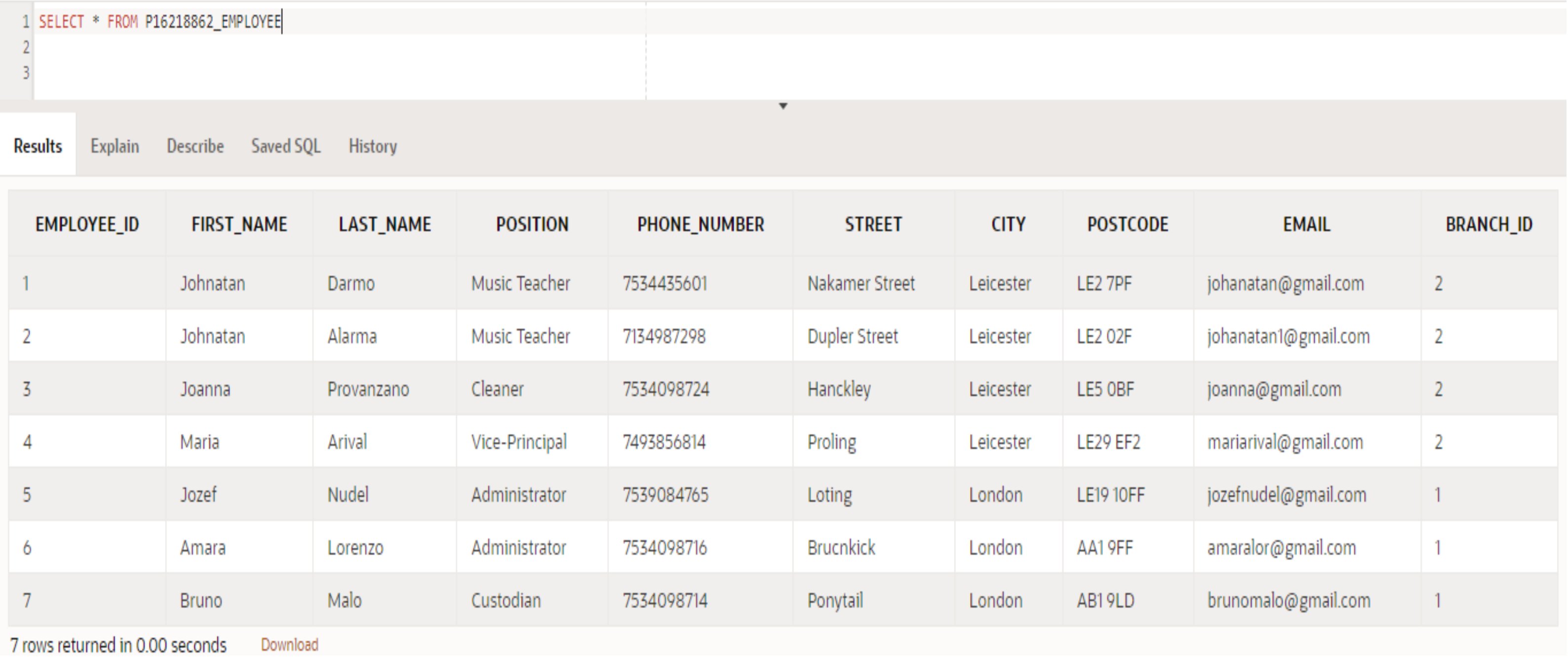
5 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_BRANCH
```

BRANCH_ID	POSTCODE	CITY	PHONE_NUMBER	EMAIL
1	LE1 2EF	Leicester	116543789	musicschoolleicester@gmail.com
2	A1 7AB	London	116534987	musicschoollondon@gmail.com

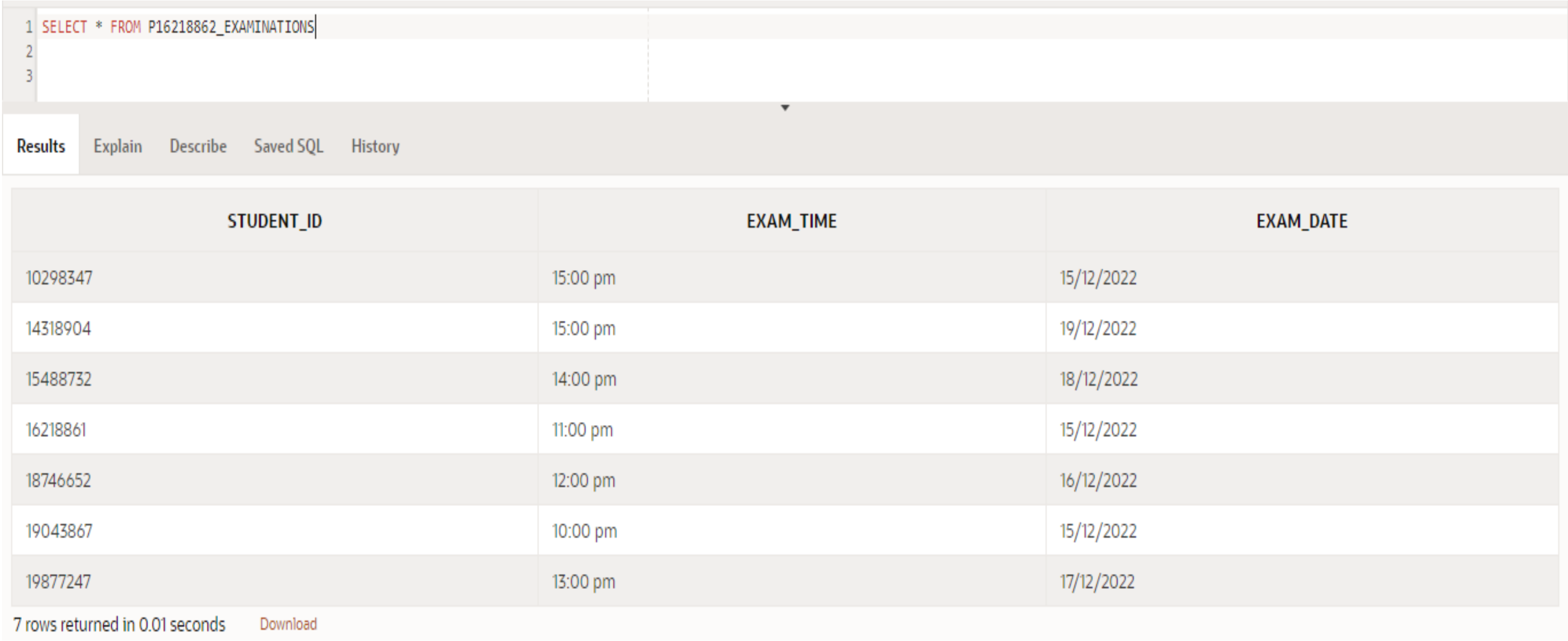
2 rows returned in 0.01 seconds



```
SELECT * FROM P16218862_EMPLOYEE
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	POSITION	PHONE_NUMBER	STREET	CITY	POSTCODE	EMAIL	BRANCH_ID
1	Johnatan	Darmo	Music Teacher	7534435601	Nakamer Street	Leicester	LE2 7PF	johanatan@gmail.com	2
2	Johnatan	Alarma	Music Teacher	7134987298	Dupler Street	Leicester	LE2 02F	johanatan1@gmail.com	2
3	Joanna	Provanzano	Cleaner	7534098724	Hanckley	Leicester	LE5 0BF	joanna@gmail.com	2
4	Maria	Arival	Vice-Principal	7493856814	Proling	Leicester	LE29 EF2	mariarival@gmail.com	2
5	Jozef	Nudel	Administrator	7539084765	Loting	London	LE19 10FF	jozefnudel@gmail.com	1
6	Amara	Lorenzo	Administrator	7534098716	Brucnkick	London	AA1 9FF	amaralor@gmail.com	1
7	Bruno	Malo	Custodian	7534098714	Ponytail	London	AB1 9LD	brunomalo@gmail.com	1

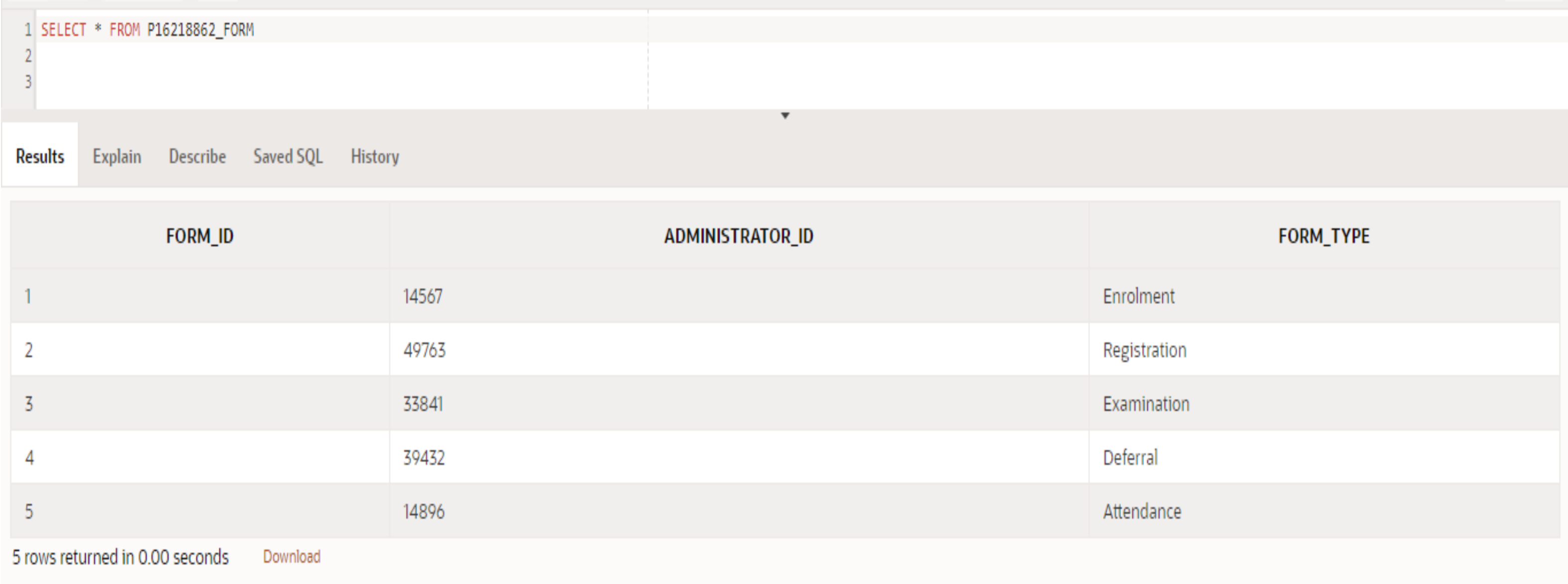
7 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_EXAMINATIONS
```

STUDENT_ID	EXAM_TIME	EXAM_DATE
10298347	15:00 pm	15/12/2022
14318904	15:00 pm	19/12/2022
15488732	14:00 pm	18/12/2022
16218861	11:00 pm	15/12/2022
18746652	12:00 pm	16/12/2022
19043867	10:00 pm	15/12/2022
19877247	13:00 pm	17/12/2022

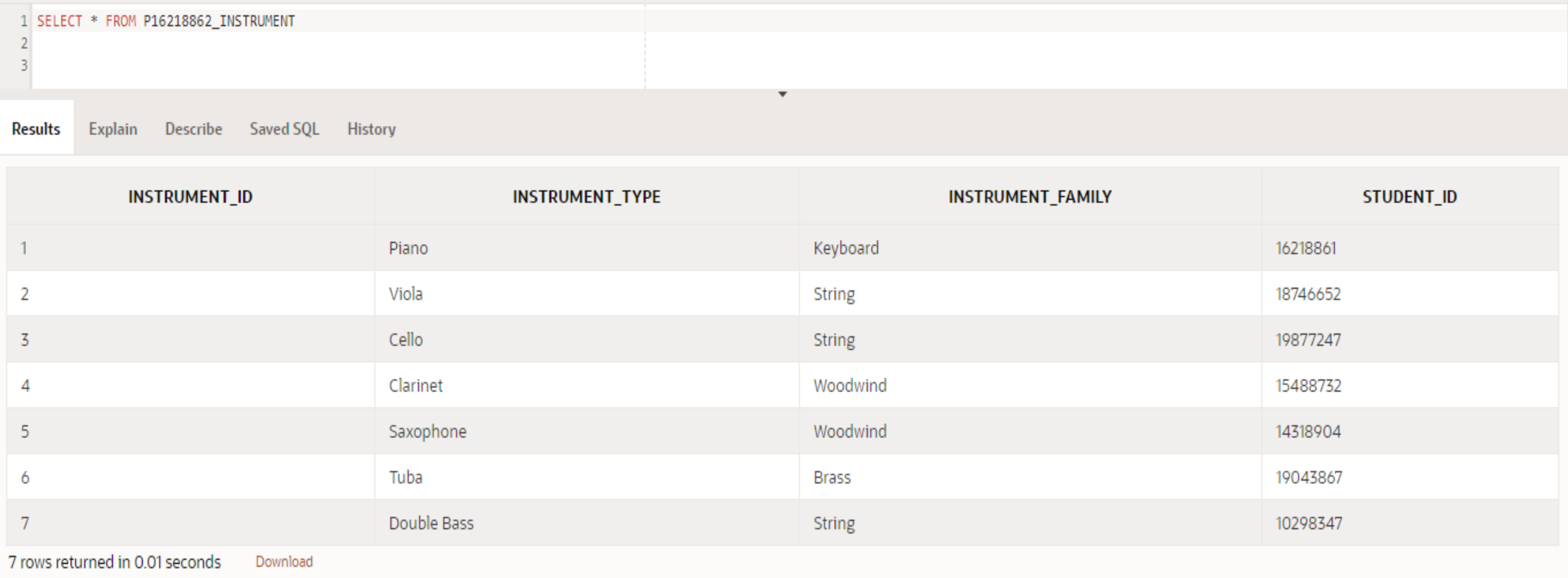
7 rows returned in 0.01 seconds



```
SELECT * FROM P16218862_FORM
```

FORM_ID	ADMINISTRATOR_ID	FORM_TYPE
1	14567	Enrolment
2	49763	Registration
3	33841	Examination
4	39432	Deferral
5	14896	Attendance

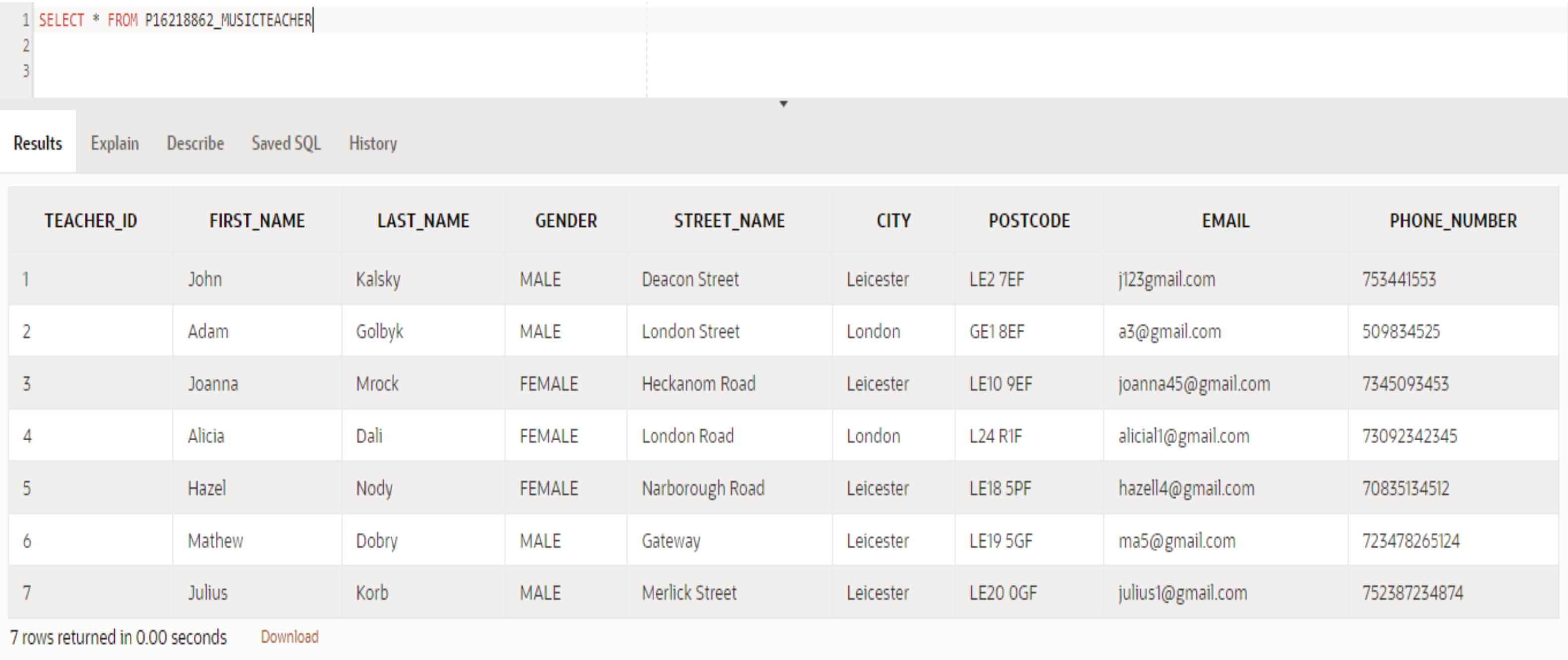
5 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_INSTRUMENT
```

INSTRUMENT_ID	INSTRUMENT_TYPE	INSTRUMENT_FAMILY	STUDENT_ID
1	Piano	Keyboard	16218861
2	Viola	String	18746652
3	Cello	String	19877247
4	Clarinet	Woodwind	15488732
5	Saxophone	Woodwind	14318904
6	Tuba	Brass	19043867
7	Double Bass	String	10298347

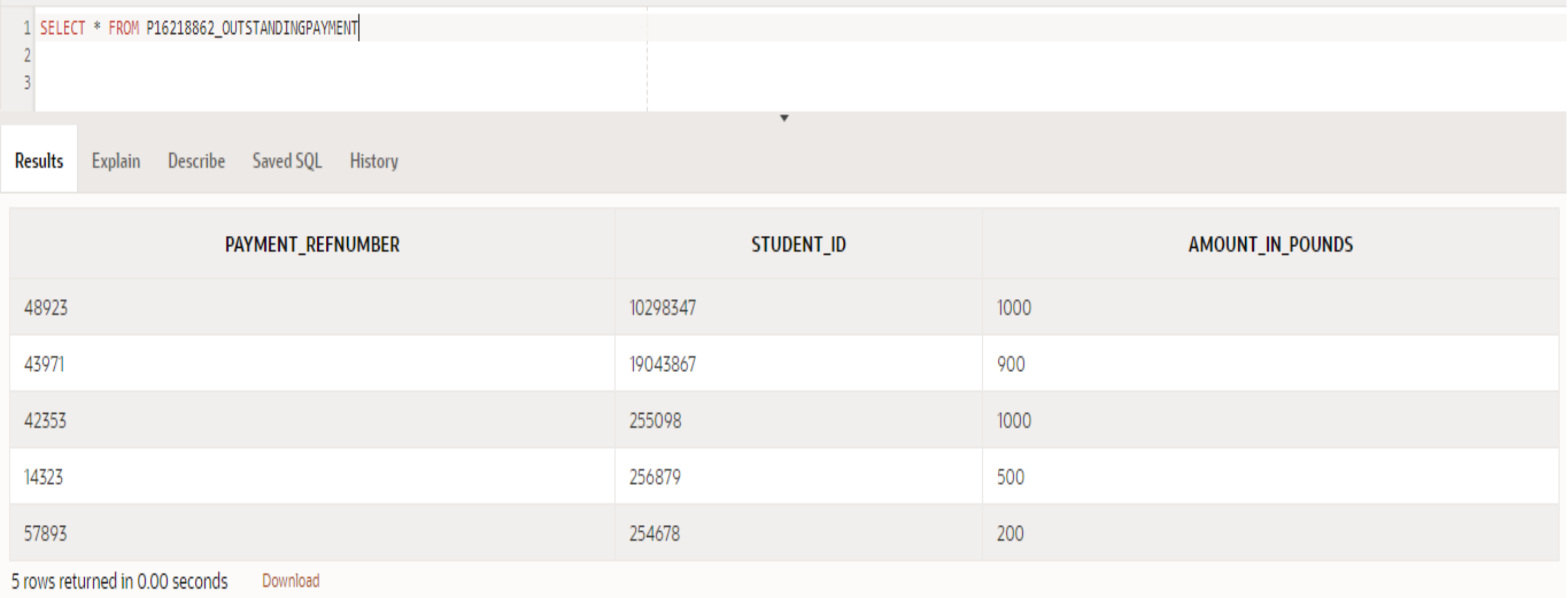
7 rows returned in 0.01 seconds



```
SELECT * FROM P16218862_MUSICTEACHER
```

TEACHER_ID	FIRST_NAME	LAST_NAME	GENDER	STREET_NAME	CITY	POSTCODE	EMAIL	PHONE_NUMBER
1	John	Kalsky	MALE	Deacon Street	Leicester	LE2 7EF	j123gmail.com	753441553
2	Adam	Golbyk	MALE	London Street	London	GE1 8EF	a3@gmail.com	509834525
3	Joanna	Mrock	FEMALE	Heckanom Road	Leicester	LE10 9EF	joanna45@gmail.com	7345093453
4	Alicia	Dali	FEMALE	London Road	London	L24 R1F	alicial1@gmail.com	73092342345
5	Hazel	Nody	FEMALE	Narborough Road	Leicester	LE18 5PF	hazell4@gmail.com	70835134512
6	Mathew	Dobry	MALE	Gateway	Leicester	LE19 5GF	ma5@gmail.com	723478265124
7	Julius	Korb	MALE	Merlick Street	Leicester	LE20 0GF	julius1@gmail.com	752387234874

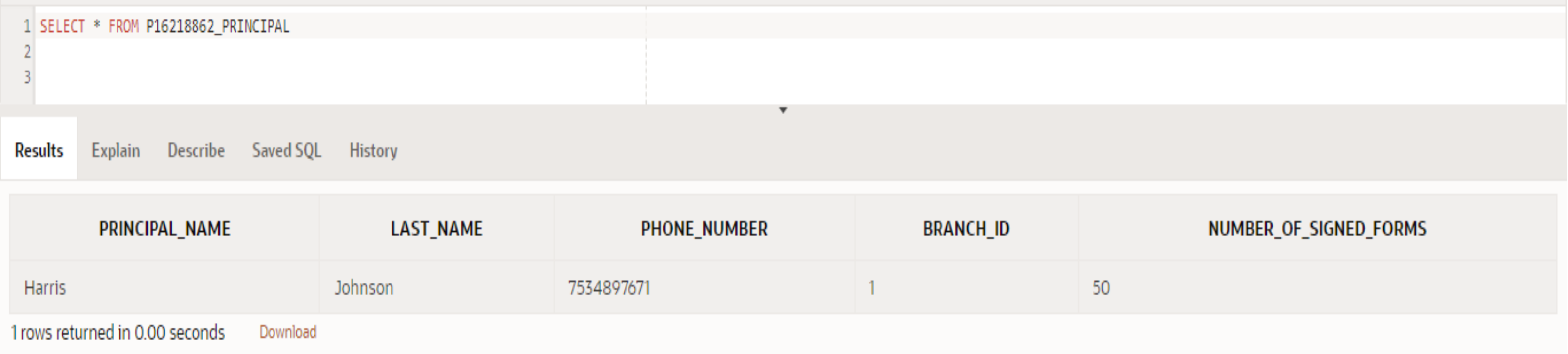
7 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_OUTSTANDINGPAYMENT
```

PAYMENT_REFNUMBER	STUDENT_ID	AMOUNT_IN_POUNDS
48923	10298347	1000
43971	19043867	900
42353	255098	1000
14323	256879	500
57893	254678	200

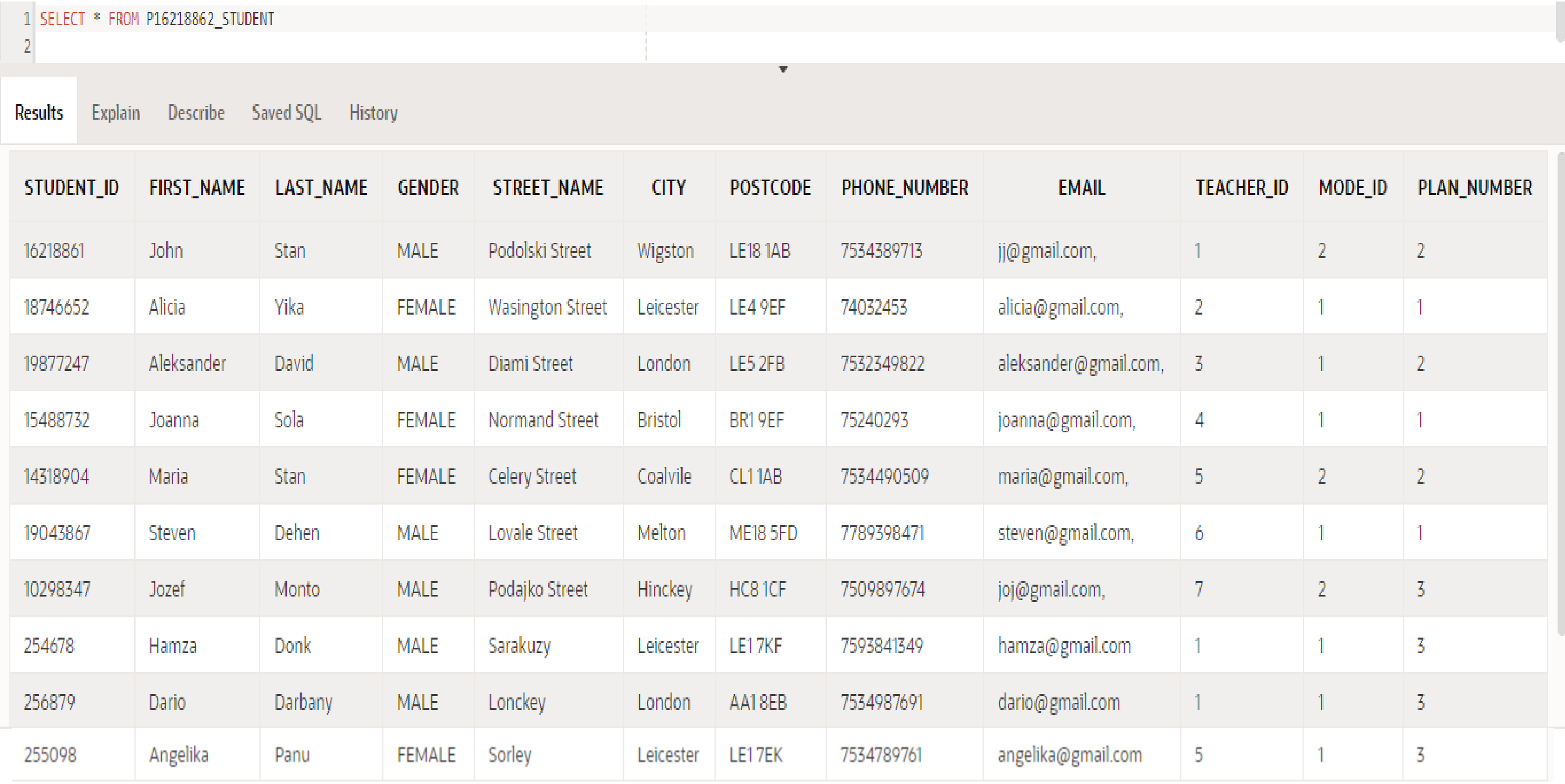
5 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_PRINCIPAL
```

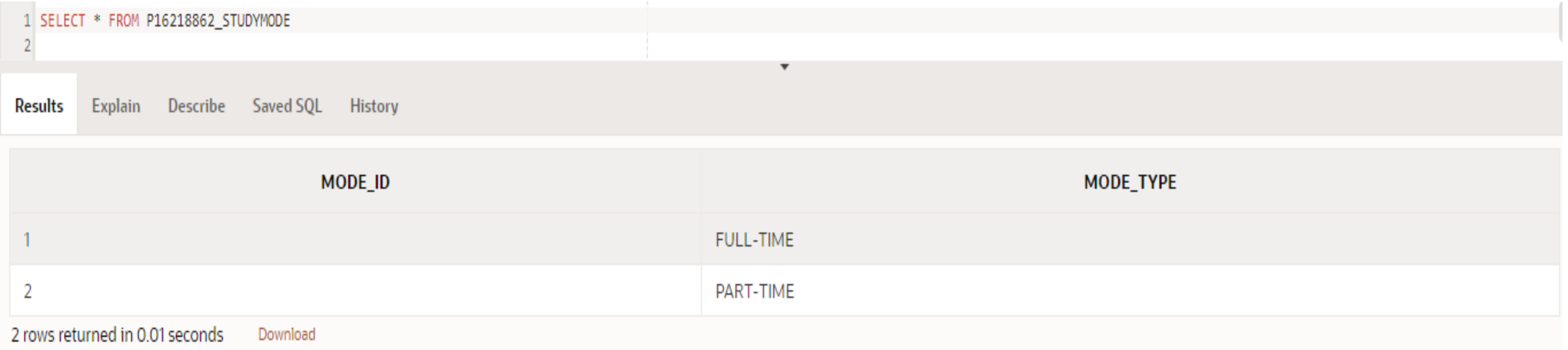
PRINCIPAL_NAME	LAST_NAME	PHONE_NUMBER	BRANCH_ID	NUMBER_OF_SIGNED_FORMS
Harris	Johnson	7534897671	1	50

1 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_STUDENT
```

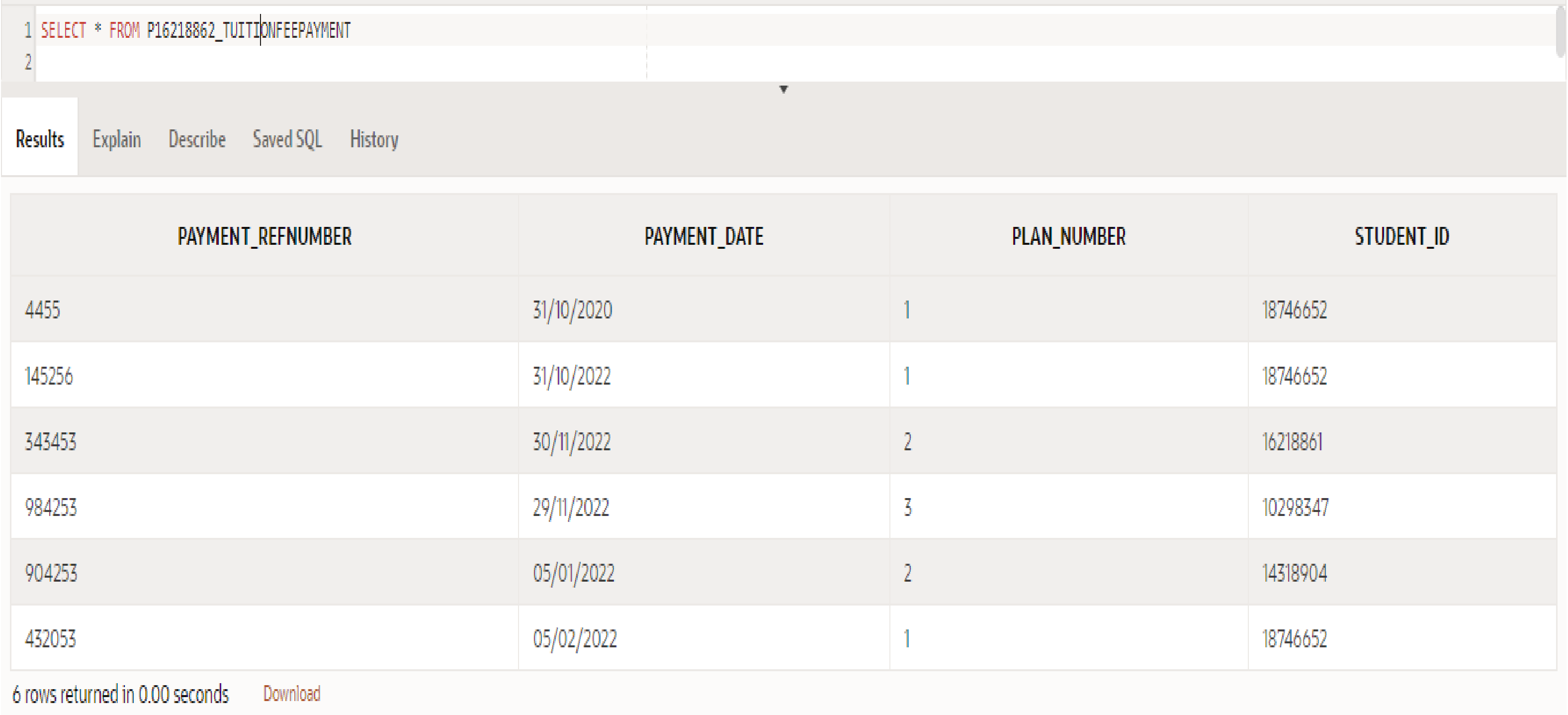
STUDENT_ID	FIRST_NAME	LAST_NAME	GENDER	STREET_NAME	CITY	POSTCODE	PHONE_NUMBER	EMAIL	TEACHER_ID	MODE_ID	PLAN_NUMBER
16218861	John	Stan	MALE	Podolski Street	Wigston	LE18 1AB	7534389713	jj@gmail.com,	1	2	2
18746652	Alicia	Yika	FEMALE	Wasington Street	Leicester	LE4 9EF	74032453	alicia@gmail.com,	2	1	1
19877247	Aleksander	David	MALE	Diami Street	London	LE5 2FB	7532349822	aleksander@gmail.com,	3	1	2
15488732	Joanna	Sola	FEMALE	Normand Street	Bristol	BR1 9EF	75240293	joanna@gmail.com,	4	1	1
14318904	Maria	Stan	FEMALE	Celery Street	Coalvile	CL1 1AB	7534490509	maria@gmail.com,	5	2	2
19043867	Steven	Dehen	MALE	Lovale Street	Melton	ME18 5FD	7789398471	steven@gmail.com,	6	1	1
10298347	Jozef	Monto	MALE	Podajko Street	Hinckey	HC8 1CF	7509897674	joj@gmail.com,	7	2	3
254678	Hamza	Donk	MALE	Sarakuzy	Leicester	LE1 7KF	7593841349	hamza@gmail.com	1	1	3
256879	Dario	Darbany	MALE	Lonckey	London	AA1 8EB	7534987691	dario@gmail.com	1	1	3
255098	Angelika	Panu	FEMALE	Sorley	Leicester	LE1 7EK	7534789761	angelika@gmail.com	5	1	3



```
SELECT * FROM P16218862_STUDYMODE
```

MODE_ID	MODE_TYPE
1	FULL-TIME
2	PART-TIME

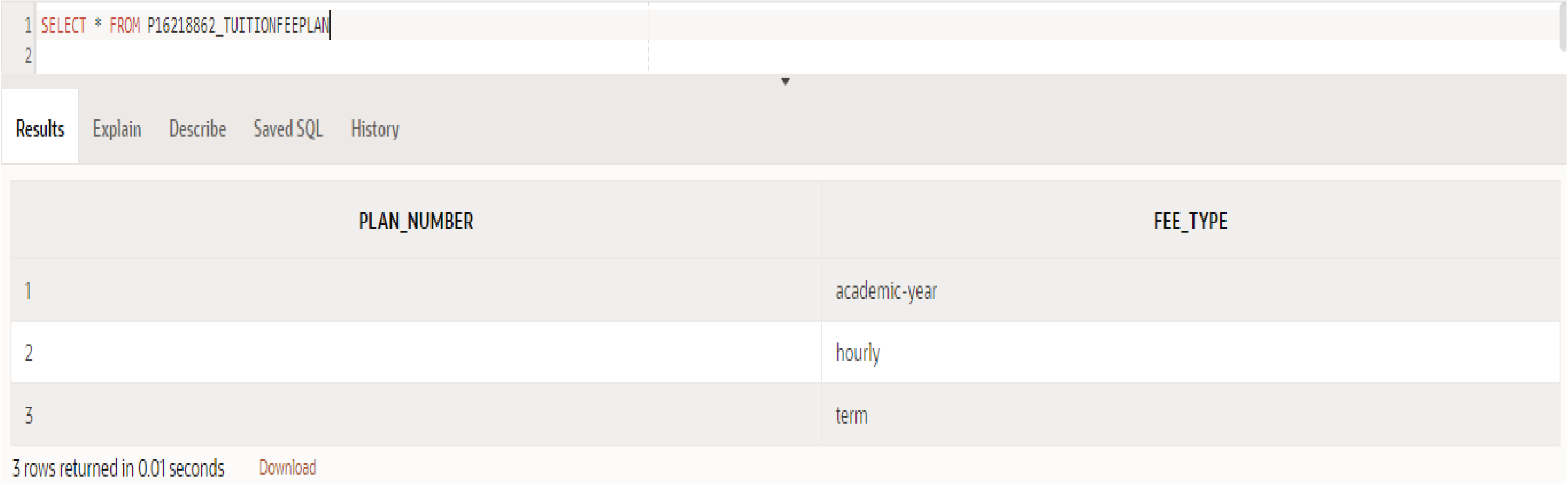
2 rows returned in 0.01 seconds



```
SELECT * FROM P16218862_TUITIONFEEPAYMENT
```

PAYMENT_REFNUMBER	PAYMENT_DATE	PLAN_NUMBER	STUDENT_ID
4455	31/10/2020	1	18746652
145256	31/10/2022	1	18746652
343453	30/11/2022	2	16218861
984253	29/11/2022	3	10298347
904253	05/01/2022	2	14318904
432053	05/02/2022	1	18746652

6 rows returned in 0.00 seconds



```
SELECT * FROM P16218862_TUITIONFEEPLAN
```

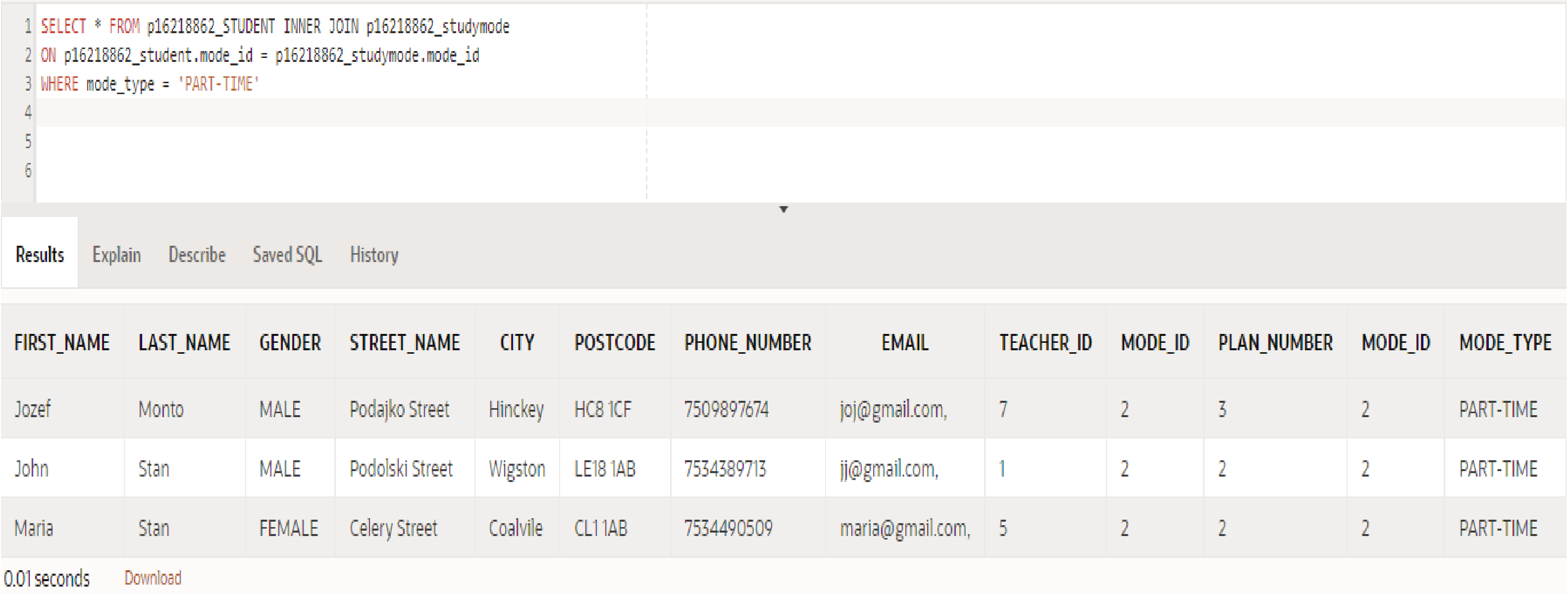
PLAN_NUMBER	FEE_TYPE
1	academic-year
2	hourly
3	term

3 rows returned in 0.01 seconds

TASK 7: SQL QUERIES

The following screenshots present six queries run in Oracle.

1. This query helps to retreivew the number of students enrolled part time and teachers associated with those students. It also gives us an information about the tuition fee plan (plan number) that the a student has chosen. Without the command in the third line(WHERE) this would give us all of the students that are enrolled part-time and full-time. It could also be replaced with **WHERE mode_type = 'FULL-TIME'**, which would return all of the student enrolled full-time. This set of commands is useful if the school wanted to retrieve such a set of information.

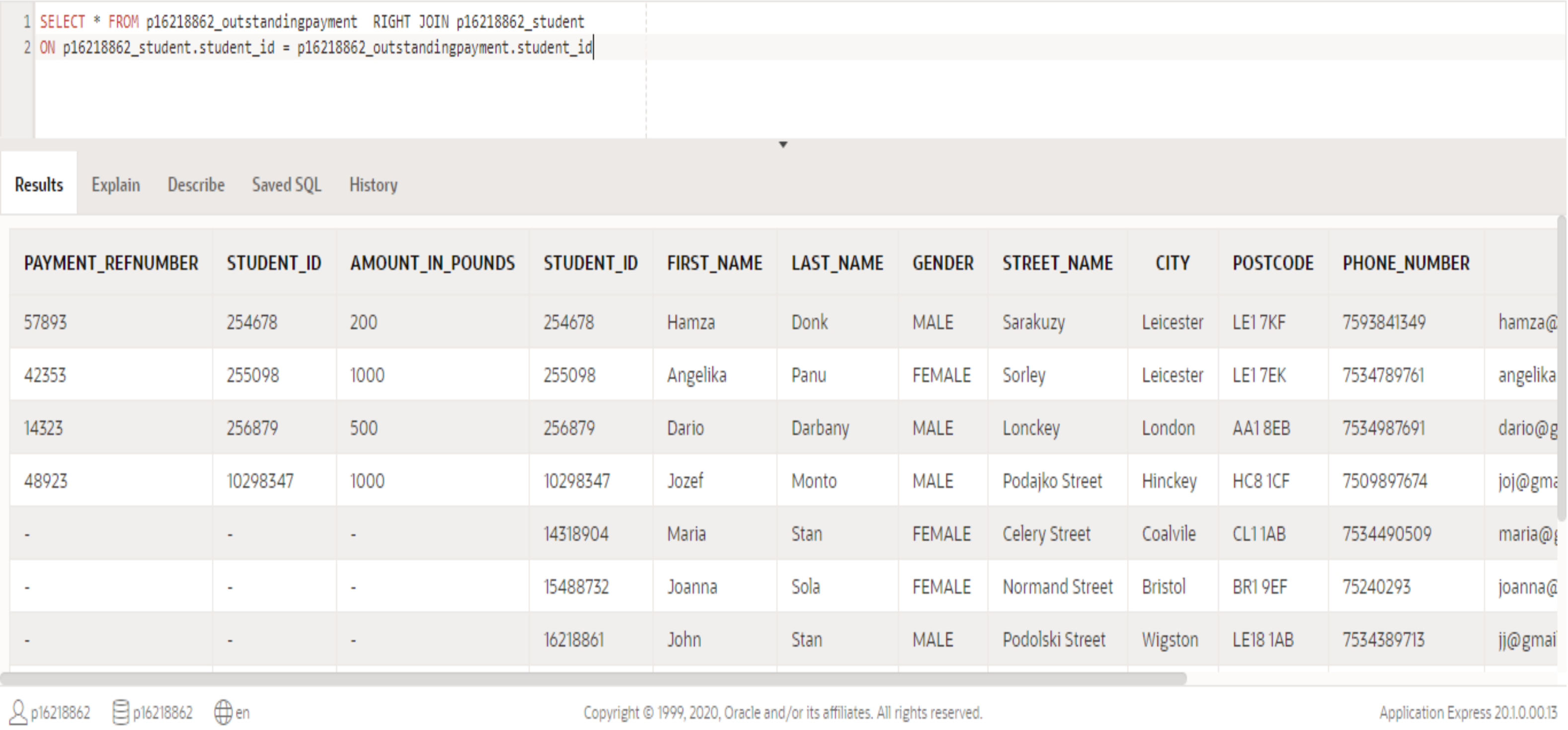


```
SELECT * FROM p16218862_STUDENT INNER JOIN p16218862_studymode
ON p16218862_student.mode_id = p16218862_studymode.mode_id
WHERE mode_type = 'PART-TIME'
```

FIRST_NAME	LAST_NAME	GENDER	STREET_NAME	CITY	POSTCODE	PHONE_NUMBER	EMAIL	TEACHER_ID	MODE_ID	PLAN_NUMBER	MODE_ID	MODE_TYPE
Jozef	Monto	MALE	Podajko Street	Hinckey	HC8 1CF	7509897674	joj@gmail.com,	7	2	3	2	PART-TIME
John	Stan	MALE	Podolski Street	Wigston	LE18 1AB	7534389713	jj@gmail.com,	1	2	2	2	PART-TIME
Maria	Stan	FEMALE	Celery Street	Coalvile	CL1 1AB	7534490509	maria@gmail.com,	5	2	2	2	PART-TIME

0.01 seconds Download

2. This query presents a way to retrieve an information about the students' outstanding payments. This information is certainly useful for the school as it's important to keep track of the number of students who paid and who didn't.

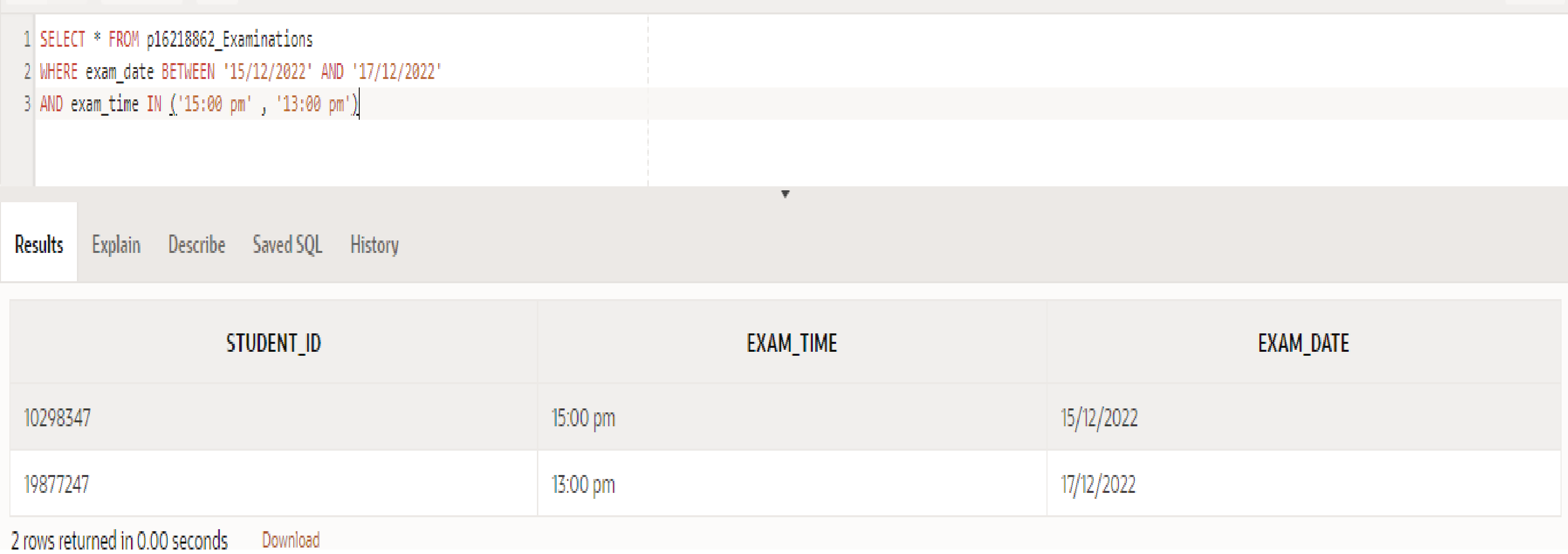


```
SELECT * FROM p16218862_outstandingpayment  RIGHT JOIN p16218862_student
ON p16218862_student.student_id = p16218862_outstandingpayment.student_id
```

PAYMENT_REFNUMBER	STUDENT_ID	AMOUNT_IN_POUNDS	STUDENT_ID	FIRST_NAME	LAST_NAME	GENDER	STREET_NAME	CITY	POSTCODE	PHONE_NUMBER	
57893	254678	200	254678	Hamza	Donk	MALE	Sarakuzy	Leicester	LE1 7KF	7593841349	hamza@
42353	255098	1000	255098	Angelika	Panu	FEMALE	Sorley	Leicester	LE1 7EK	7534789761	angelika
14323	256879	500	256879	Dario	Darbany	MALE	Lonckey	London	AA1 8EB	7534987691	dario@g
48923	10298347	1000	10298347	Jozef	Monto	MALE	Podajko Street	Hinckey	HC8 1CF	7509897674	joj@gma
-	-	-	14318904	Maria	Stan	FEMALE	Celery Street	Coalvile	CL1 1AB	7534490509	maria@
-	-	-	15488732	Joanna	Sola	FEMALE	Normand Street	Bristol	BR1 9EF	75240293	joanna@
-	-	-	16218861	John	Stan	MALE	Podolski Street	Wigston	LE18 1AB	7534389713	jj@gmai

p16218862
p16218862
en
Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.
Application Express 20.1.0.00.13

3. This query helps the school to retrieve the information about all of the students who will be taking the exams between the specific dates and even the specific time of the examination.

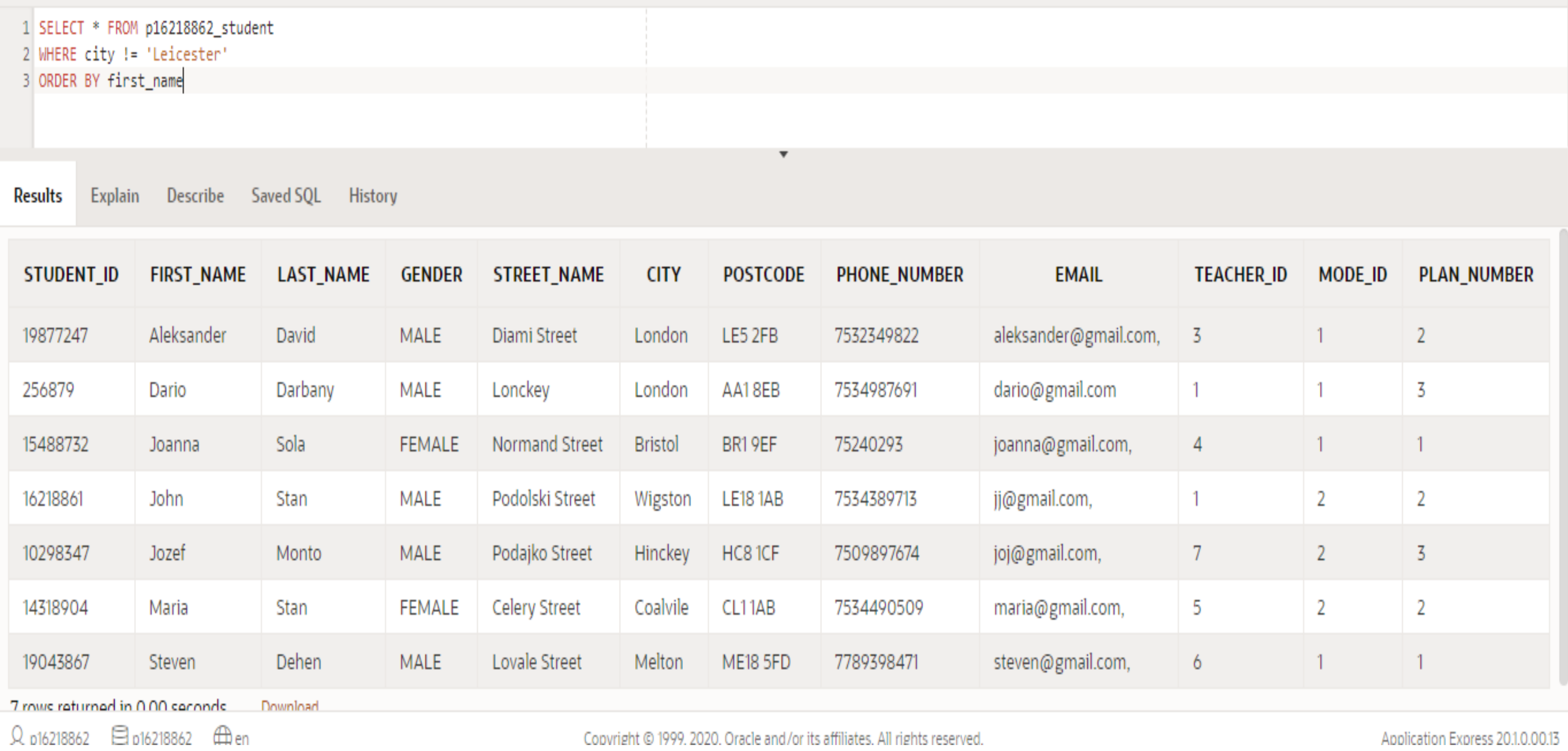


```sql
SELECT * FROM p16218862_Examinations
WHERE exam_date BETWEEN '15/12/2022' AND '17/12/2022'
AND exam_time IN ('15:00 pm' , '13:00 pm')
```

STUDENT_ID	EXAM_TIME	EXAM_DATE
10298347	15:00 pm	15/12/2022
19877247	13:00 pm	17/12/2022

2 rows returned in 0.00 seconds

4. If the school wanted to know all of the students that live outside of the branches' locations (Leicester, London) it could query the statements below.

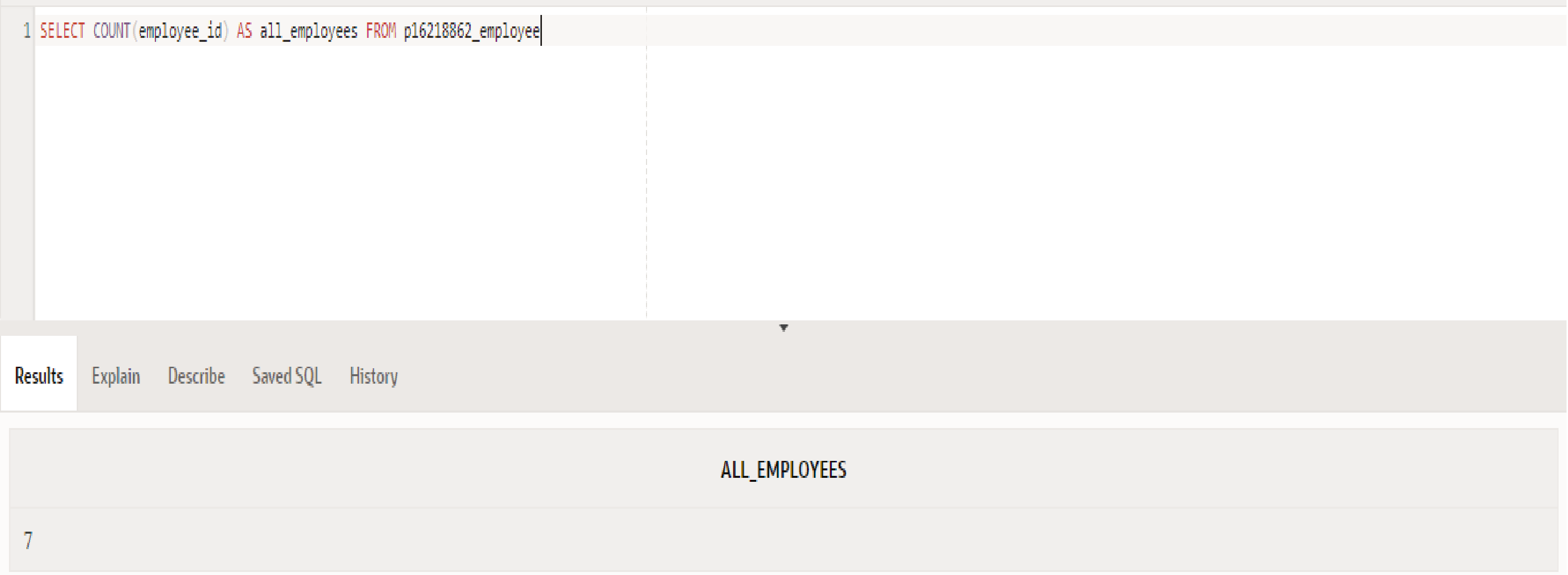


```sql
SELECT * FROM p16218862_student
WHERE city != 'Leicester'
ORDER BY first_name
```

STUDENT_ID	FIRST_NAME	LAST_NAME	GENDER	STREET_NAME	CITY	POSTCODE	PHONE_NUMBER	EMAIL	TEACHER_ID	MODE_ID	PLAN_NUMBER
19877247	Aleksander	David	MALE	Diami Street	London	LE5 2FB	7532349822	aleksander@gmail.com,	3	1	2
256879	Dario	Darbany	MALE	Lonckey	London	AA1 8EB	7534987691	dario@gmail.com	1	1	3
15488732	Joanna	Sola	FEMALE	Normand Street	Bristol	BR1 9EF	75240293	joanna@gmail.com,	4	1	1
16218861	John	Stan	MALE	Podolski Street	Wigston	LE18 1AB	7534389713	jj@gmail.com,	1	2	2
10298347	Jozef	Monto	MALE	Podajko Street	Hinckey	HC8 1CF	7509897674	joj@gmail.com,	7	2	3
14318904	Maria	Stan	FEMALE	Celery Street	Coalvile	CL1 1AB	7534490509	maria@gmail.com,	5	2	2
19043867	Steven	Dehen	MALE	Lovale Street	Melton	ME18 5FD	7789398471	steven@gmail.com,	6	1	1

7 rows returned in 0.00 seconds

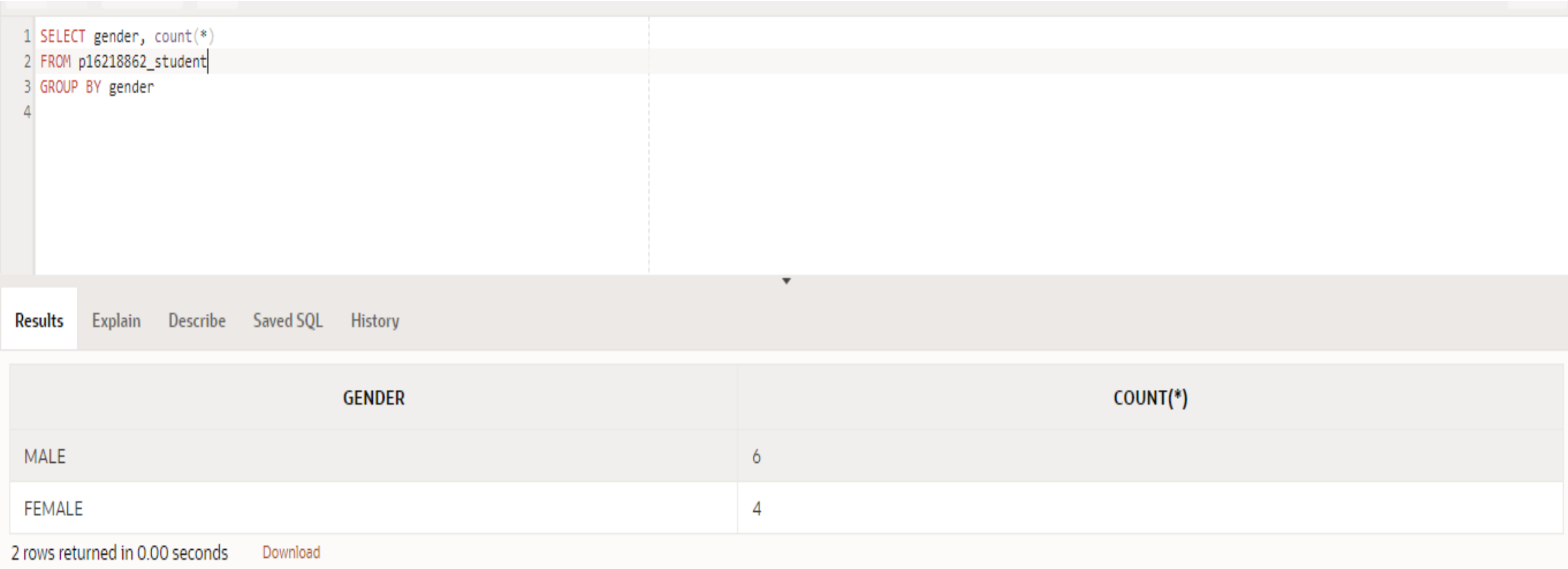
5.This query could help the school to quickly retrieve the number of employee currently hired.



```
SELECT COUNT(employee_id) AS all_employees FROM p16218862_employee
```

ALL_EMPLOYEES
7

6. This query helps to retrieve data about the number of boys and girl (men and women) currently enrolled in the school.



```
SELECT gender, count(*)
FROM p16218862_student
GROUP BY gender
```

GENDER	COUNT(*)
MALE	6
FEMALE	4

2 rows returned in 0.00 seconds

STAGE 3

TASK 8: DATA-CENTRED ARCHITECTURE

An early approach of organizing and storing data using computers was a File-Based System. File System, as the name indicates stored data as a group of files, and each file was independent from another. It is a ‘collection of application programs that perform services for end users’ (Obembe F 2021, p. 4). Every single program managed it’s own data, hence there were quite a few limitations to this system. One of the biggest are:

- Separated and Isolated Data – data held in different places
- Duplication of Data – overlapping data in two or more different places
- Data Security – data held in a flat file, which is easily accessible

In order to fix or at least minimize all of the issues of the file-based system there was a need for a new data system to be developed.

This is why a data-centred or database approach arose and resulted in creating a DBMS, which stands for Database Management System. A database is simply a place where the data is stored. If we have a database on our computer, and we put some type of data into it, and then open our computer after a few hours, days or even months and years, the data that we inserted will still be there. This is the basic and most important factor of the database. Information in the database is stored in logically related tables with attributes and records (columns and rows). Those contain information that are suited to ones’ needs; a company, organization etc. DBMS is a software system that creates and controls access to the database and interacts with the database through SQL (structured query language).

Large organizations and even smaller companies always require a persistent data storage. All of the information about a specific group of people and their actions needs to be created and maintained in such a way so that there is no place for wasting too much time, the data is accurate and secure, but at the same time easily updated by appropriate administrative units. One of the main advantages of the database (data-centred) approach are ‘controlled access to the database, a security system, an integrity system, a concurrency control system, a recovery control system, a user-accessible catalog’ (Obembe F 2021). Most likely the most important underlying factor of the DBMS is that it is simple. DBMS presents us with a simple to understand representation of stored information. It creates a place for users where data can be easily managed. This also provides a way for a quick reaction to any important changes. The integrity of the system is well maintained because we have available number of tools that help to eliminate or reduce the redundant information. For example RDBMS (Relational database management system which is an advanced version of DBMS) allows us to take the database through the process of Normalization – a set of well defined rules to reduce redundant data and to make easier for the users to maintain it).

STAGE 4

TASK 9: DATA MODELS

A data model is model that represents the real-world entities. It shows us how those entities are connected to one another and their relationships.

There are several data models. One of the most common are:

- Conceptualized data model
- Logical data model
- Physical data model

Those are used in a process of designing a database. Each model contains more detailed and user/ organization focus information than the previous. Here, all of the models can be thought of as steps to create database. At the stage of a conceptualized design the most important entities and relationships between them are drawn. At this stage a person designing and developing a database will think of the most important factors of the company that the database is being develop for and present them as entities using a diagram. The first requirements for the project are defined, created and joined on the diagram. This is achieved by using an ERD (Entity relationship diagram) and later by EER (Enhanced entity relationship diagram). Logical design builds up on the conceptualized model and adds more details such as columns. Logical design can be useful while developing a data warehouse project. Physical data model is a blueprint and a roadmap for the relational database. Entities and their detailed relationships are presented here with all of their primary and foreign keys. At this stage detailed diagram is presented with attributes and their length , the data type that they accepts and so on.

There are also other types of data models:

1.Record-based data model

2.Object-oriented data model

3.Object-relational data model

Record based data model is a part of first generation systems. It is further divided into three data models:

- Hierarchical
- Network
- Relational

Record based model presents us with a database that has fixed set of records. It’s subset – hierarchical model was introduced by IBM in 1966. It has a reversed tree-like format and represent the collection of records. It’s main advantage was that it was simple in it’s presentation, but at the same time it wasn’t very flexible. Network data model also present us with the collection of records but it provides a way for one segment (record) of data to have more than one parent. The independence of data and it’s integrity were one of the main advantages of this model, but the complexity of the system didn’t allow for a lot of flexibility. Relational data model are still widely used, because it’s much more flexible and easier to implement than its’ predecessors. Files can be related to one another by a common attribute, which wasn’t possible in the network data model. Relational data model allow for a relational database to be developed, which normally use SQL – an easy to use language that allows to send queries to the database.

BIBLIOGRAPHY

Connolly, T.M. and Begg, C.E. (2015). *Database systems : a practical approach to design, implementation, and management*. Harlow, Essex, England: Pearson Education Limited.

Obembe, F 2021, *Workshop 1: Database Systems and Designs, lecture notes*, Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 2: Data models and Introduction to SQL, lectures notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 3: Relational data model – Normalization, lecture notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 4: Entity Relationship Modelling, lectures notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 5: (Enhanced) Entity Relationship Modelling, lectures notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 6: Design Methodology 1 & more SQL, lectures notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 7: Design Methodology 2 & more SQL, lectures notes,* Database Systems and Design, De Montfort University

Obembe, F 2021, *Workshop 9: Other data models and data warehousing, lectures notes,* Database Systems and Design, De Montfort University