North South University

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Assignment on

ORDBMS & NoSQL Database Design, Implementation and Query

Course Information
Advanced Database Systems
CSE411 (Section 2)
SPRING 2025

Submitted by
Saif Mohammed 2121913042

Submitted to
Dr. Abu Sayed Md. Latiful Hoque
Professor, CSE, BUET
&
Visiting Professor, ECE, NSU

Submission Date
14 April, 2025

CSE411 Assignment

Contents

1 Part-1: ORDBMS

1.1 Task-1: Define types and inheritance

```
-- Define Types

CREATE OR REPLACE TYPE Name_Type AS OBJECT (
   FirstName VARCHAR2(50),
   MiddleName VARCHAR2(50),
   LastName VARCHAR2(50)
);

CREATE OR REPLACE TYPE Address_Detail_Type AS OBJECT (
   HouseNumber VARCHAR2(30),
   Street VARCHAR2(100),
   Thana VARCHAR2(50),
   District VARCHAR2(50)
);

CREATE OR REPLACE TYPE Address_Type AS OBJECT (
   PresentAddr Address_Detail_Type,
   PermanentAddr Address_Detail_Type
);

CREATE OR REPLACE TYPE Educational_Qualifiaction_Type AS OBJECT (
   Degree VARCHAR2(100),
   Institution VARCHAR2(150),
   Grad_Year NUMBER(4)
);

CREATE OR REPLACE TYPE Player_Type AS OBJECT (
   Game VARCHAR2(50),
   Score NUMBER
);

CREATE OR REPLACE TYPE Organizer_Type AS OBJECT (
   Club_Name VARCHAR2(100),
   Start_Date DATE,
   End_Date DATE
);

CREATE OR REPLACE TYPE FamilyTree_Type AS OBJECT (
   Family_ID NUMBER,
   Family_Name Name_Type,
   DOB DATE
);

CREATE OR REPLACE TYPE Phone_Number_List AS VARRAY(10) OF VARCHAR2(20);
CREATE OR REPLACE TYPE Email_Address_List AS VARRAY(10) OF VARCHAR2(100);
CREATE OR REPLACE TYPE Research_Interest_List AS VARRAY(20) OF VARCHAR2
    (100);
CREATE OR REPLACE TYPE Programming_Language_List AS VARRAY(20) OF VARCHAR2
    (50);
CREATE OR REPLACE TYPE Educational_Qualifications AS TABLE OF
    Educational_Qualifiaction_Type;
CREATE OR REPLACE TYPE Player_Roles_List AS VARRAY(5) OF Player_Type;
CREATE OR REPLACE TYPE Organizer_Roles_List AS VARRAY(5) OF Organizer_Type;
CREATE OR REPLACE TYPE Student_Type AS OBJECT (
   ID NUMBER,
   Student_Name Name_Type,
   DOB DATE,
   CGPA NUMBER(3,2),
   Total_Credits NUMBER(3),
   Department VARCHAR2(100),
   Addresses Address_Type,
   Phone_Numbers Phone_Number_List,
   Email_Addresses Email_Address_List,
   Research_Interests Research_Interest_List,
   Prog_Knowledge Programming_Language_List,
   Educations Educational_Qualifications,
   Father REF FamilyTree_Type,
   Father_of_Father REF FamilyTree_Type,
   Mother REF FamilyTree_Type,
   Mother_of_Mother REF FamilyTree_Type,
   Player_Roles Player_Roles_List,
   Organizer_Roles Organizer_Roles_List
);
```

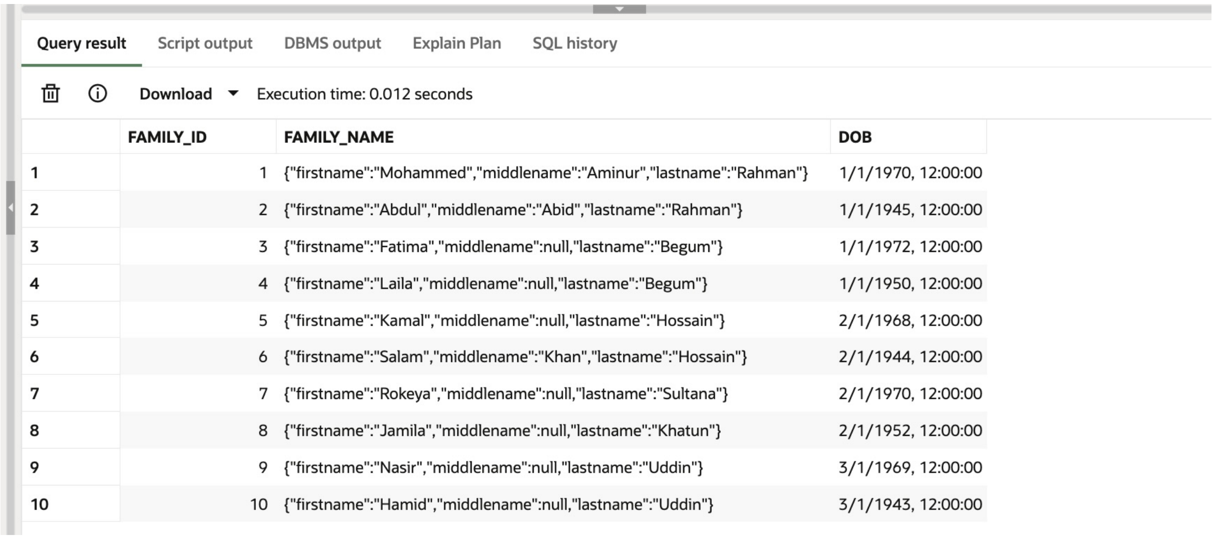
1.2 Task-2: Create tables using the types

```
-- TABLE CREATION

CREATE TABLE FamilyTrees OF FamilyTree_Type (
   Family_ID PRIMARY KEY,
   CONSTRAINT ppl_name_nn CHECK (
      Family_Name.FirstName IS NOT NULL AND Family_Name.LastName IS NOT
          NULL
   )
);

CREATE TABLE Students OF Student_Type (
   ID PRIMARY KEY,
   Student_Name NOT NULL,
   Department NOT NULL,
   Addresses NOT NULL,
   Phone_Numbers NOT NULL,
   Email_Addresses NOT NULL,
   SCOPE FOR (Father) IS FamilyTrees,
   SCOPE FOR (Father_of_Father) IS FamilyTrees,
   SCOPE FOR (Mother) IS FamilyTrees,
   SCOPE FOR (Mother_of_Mother) IS FamilyTrees
) NESTED TABLE Educations STORE AS Educational_Qualification_List;
```

1.3 Task-3: Insert sample data

```
-- insertion in FamilyTrees Table

INSERT INTO FamilyTrees VALUES (1, Name_Type(’Mohammed’, ’Aminur’, ’Rahman’
   ), DATE ’1970-01-01’);
INSERT INTO FamilyTrees VALUES (2, Name_Type(’Abdul’, ’Abid’, ’Rahman’),
   DATE ’1945-01-01’);
```

```
INSERT INTO FamilyTrees VALUES (3, Name_Type('Fatima', NULL, 'Begum'), DATE
     '1972-01-01');
INSERT INTO FamilyTrees VALUES (4, Name_Type('Laila', NULL, 'Begum'), DATE
    '1950-01-01');
INSERT INTO FamilyTrees VALUES (5, Name_Type('Kamal', NULL, 'Hossain'),
    DATE '1968-02-01');
INSERT INTO FamilyTrees VALUES (6, Name_Type('Salam', 'Khan', 'Hossain'),
    DATE '1944-02-01');
INSERT INTO FamilyTrees VALUES (7, Name_Type('Rokeya', NULL, 'Sultana'),
    DATE '1970-02-01');
INSERT INTO FamilyTrees VALUES (8, Name_Type('Jamila', NULL, 'Khatun'),
    DATE '1952-02-01');
INSERT INTO FamilyTrees VALUES (9, Name_Type('Nasir', NULL, 'Uddin'), DATE
    '1969-03-01');
INSERT INTO FamilyTrees VALUES (10, Name_Type('Hamid', NULL, 'Uddin'), DATE
     '1943-03-01');
```

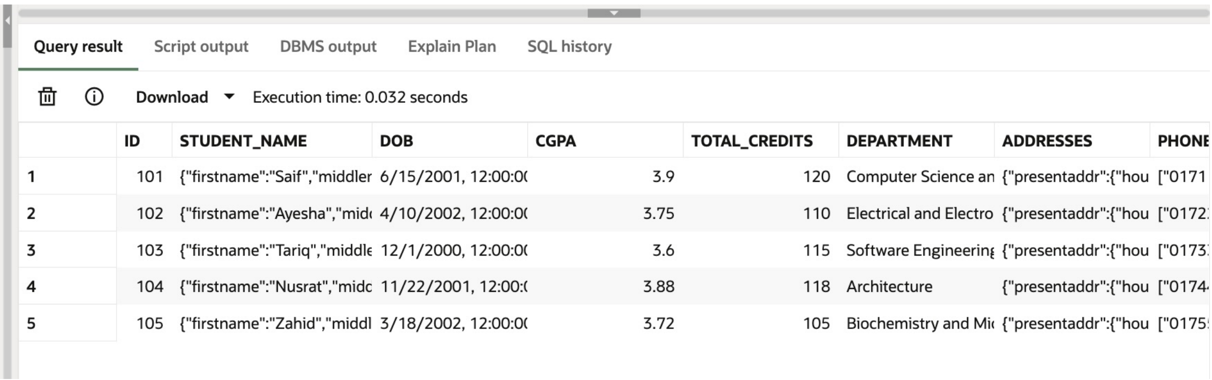


Execution time: 0.012 seconds

	FAMILY_ID	FAMILY_NAME	DOB
1	1	{"firstname":"Mohammed","middlename":"Aminur","lastname":"Rahman"}	1/1/1970, 12:00:00
2	2	{"firstname":"Abdul","middlename":"Abid","lastname":"Rahman"}	1/1/1945, 12:00:00
3	3	{"firstname":"Fatima","middlename":null,"lastname":"Begum"}	1/1/1972, 12:00:00
4	4	{"firstname":"Laila","middlename":null,"lastname":"Begum"}	1/1/1950, 12:00:00
5	5	{"firstname":"Kamal","middlename":null,"lastname":"Hossain"}	2/1/1968, 12:00:00
6	6	{"firstname":"Salam","middlename":"Khan","lastname":"Hossain"}	2/1/1944, 12:00:00
7	7	{"firstname":"Rokeya","middlename":null,"lastname":"Sultana"}	2/1/1970, 12:00:00
8	8	{"firstname":"Jamila","middlename":null,"lastname":"Khatun"}	2/1/1952, 12:00:00
9	9	{"firstname":"Nasir","middlename":null,"lastname":"Uddin"}	3/1/1969, 12:00:00
10	10	{"firstname":"Hamid","middlename":null,"lastname":"Uddin"}	3/1/1943, 12:00:00

Figure 1: FamilyTrees Table

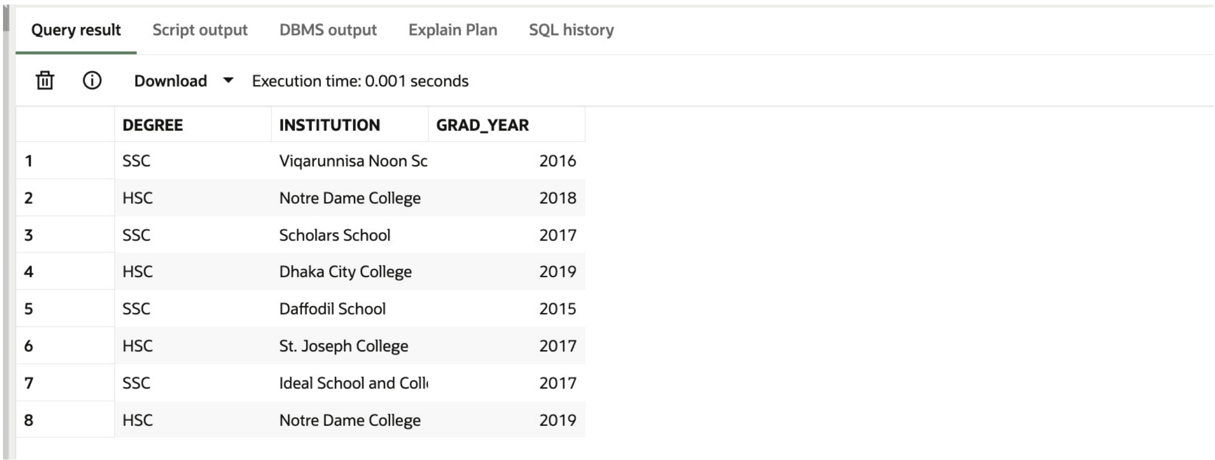
```
-- insertion in Students Table

INSERT INTO Students VALUES (
   101,
   Name_Type('Saif', 'Mohamed', 'Rahman'),
   DATE '2001-06-15',
   3.90,
   120,
   'Computer Science and Engineering',
   Address_Type(
      Address_Detail_Type('H-12', 'Road-10', 'Banasree', 'Dhaka'),
      Address_Detail_Type('H-22', 'Road-2', 'Rampura', 'Dhaka')
   ),
   Phone_Number_List('01711111111', '01611111111'),
   Email_Address_List('saif.rahman@northsouth.edu', 'saif.personal@gmail.
       com'),
   Research_Interest_List('Deep Learning', 'Natural Language Processing', '
       Computer Vision'),
   Programming_Language_List('Python', 'C++', 'Java'),
   Educational_Qualifications(
      Educational_Qualifiaction_Type(’SSC’, ’Viqarunnisa Noon School’,
         2016),
      Educational_Qualifiaction_Type(’HSC’, ’Notre Dame College’, 2018)
   ),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 1),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 2),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 3),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 4),
   Player_Roles_List(Player_Type(’Football’, 85), Player_Type(’Table Tennis
      ’, 90)),
   Organizer_Roles_List(Organizer_Type(’NSU ACM’, DATE ’2022-01-01’, DATE ’
      2023-01-01’))
);


INSERT INTO Students VALUES (
   102,
   Name_Type(’Ayesha’, NULL, ’Hossain’),
   DATE ’2002-04-10’,
   3.75,
   110,
   ’Electrical and Electronic Engineering’,
   Address_Type(
      Address_Detail_Type(’H-34’, ’Road-5’, ’Uttara’, ’Dhaka’),
      Address_Detail_Type(’H-12’, ’Road-9’, ’Dhanmondi’, ’Dhaka’)
   ),
   Phone_Number_List(’01722222222’),
   Email_Address_List(’ayesha.hossain@northsouth.edu’),
   NULL,
   Programming_Language_List(’C’, ’Python’, ’MATLAB’),
   Educational_Qualifications(
      Educational_Qualifiaction_Type(’SSC’, ’Scholars School’, 2017),
      Educational_Qualifiaction_Type(’HSC’, ’Dhaka City College’, 2019)
   ),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 5),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 6),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 7),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 8),
   Player_Roles_List(Player_Type(’Chess’, 70)),
   Organizer_Roles_List(Organizer_Type(’IEEE NSU SB’, DATE ’2021-03-01’,
      DATE ’2022-02-01’))
);


INSERT INTO Students VALUES (
   103,
   Name_Type(’Tariq’, ’Hasan’, ’Uddin’),
   DATE ’2000-12-01’,
   3.60,
   115,
   ’Software Engineering’,
   Address_Type(
      Address_Detail_Type(’H-8’, ’Mirpur 10’, ’Mirpur’, ’Dhaka’),
      Address_Detail_Type(’H-2’, ’Shyamoli’, ’Mohammadpur’, ’Dhaka’)
   ),
   Phone_Number_List(’01733333333’),
   Email_Address_List(’tariq.hasan@northsouth.edu’),
   Research_Interest_List(’Web Development’, ’Agile Methodologies’),
   NULL,
   Educational_Qualifications(
      Educational_Qualifiaction_Type(’SSC’, ’Daffodil School’, 2015),
      Educational_Qualifiaction_Type(’HSC’, ’St. Joseph College’, 2017)
   ),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 9),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 10),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 3),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 4),
   Player_Roles_List(Player_Type(’Basketball’, 88)),
   NULL
);


INSERT INTO Students VALUES (
   104,
   Name_Type(’Nusrat’, ’Jahan’, ’Rimi’),
   DATE ’2001-11-22’,
   3.88,
   118,
   ’Architecture’,
   Address_Type(
      Address_Detail_Type(’H-44’, ’New Eskaton’, ’Ramna’, ’Dhaka’),
      Address_Detail_Type(’H-21’, ’Road-13’, ’Mohakhali’, ’Dhaka’)
   ),
   Phone_Number_List(’01744444444’),
   Email_Address_List(’nusrat.jahan@northsouth.edu’, ’nusrat.jr@gmail.com’)
      ,
   Research_Interest_List(’Urban Planning’, ’Sustainable Design’, ’3D
      Modeling’),
   Programming_Language_List(’AutoCAD’, ’SketchUp’, ’Python’),
   NULL,
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 1),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 2),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 3),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 4),
   NULL,
   Organizer_Roles_List(Organizer_Type(’NSU Earth Club’, DATE ’2022-01-15’,
       DATE ’2023-01-15’))
);


INSERT INTO Students VALUES (
   105,
   Name_Type(’Zahid’, ’Imtiaz’, ’Khan’),
   DATE ’2002-03-18’,
   3.72,
   105,
   ’Biochemistry and Microbiology’,
   Address_Type(
      Address_Detail_Type(’H-55’, ’Road-8’, ’Bashundhara’, ’Dhaka’),
      Address_Detail_Type(’H-31’, ’Lane-4’, ’Malibagh’, ’Dhaka’)
   ),
   Phone_Number_List(’01755555555’),
   Email_Address_List(’zahid.khan@northsouth.edu’, ’zahid.bio@gmail.com’),
   Research_Interest_List(’Genetics’, ’Immunology’, ’Biotech Innovations’),
   Programming_Language_List(’R’, ’Python’, ’SQL’),
   Educational_Qualifications(
      Educational_Qualifiaction_Type('SSC', 'Ideal School and College',
          2017),
      Educational_Qualifiaction_Type('HSC', 'Notre Dame College', 2019)
   ),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 5),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 6),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 7),
   (SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 8),
   NULL,
   Organizer_Roles_List(Organizer_Type('NSU Biology Club', DATE '2023-01-01
      ', DATE '2024-01-01'))
);
```



Execution time: 0.032 seconds

	ID	STUDENT_NAME	DOB	CGPA	TOTAL_CREDITS	DEPARTMENT	ADDRESSES	PHONE
1	101	{"firstname":"Saif","middler	6/15/2001, 12:00:0(	3.9	120	Computer Science an	{"presentaddr":{"hou	["0171
2	102	{"firstname":"Ayesha","mid	4/10/2002, 12:00:0(	3.75	110	Electrical and Electro	{"presentaddr":{"hou	["0172
3	103	{"firstname":"Tariq","middle	12/1/2000, 12:00:0(	3.6	115	Software Engineering	{"presentaddr":{"hou	["0173
4	104	{"firstname":"Nusrat","midc	11/22/2001, 12:00:(	3.88	118	Architecture	{"presentaddr":{"hou	["0174
5	105	{"firstname":"Zahid","middl	3/18/2002, 12:00:0(	3.72	105	Biochemistry and Mi	{"presentaddr":{"hou	["0175

Figure 2: Students Table



Execution time: 0.001 seconds

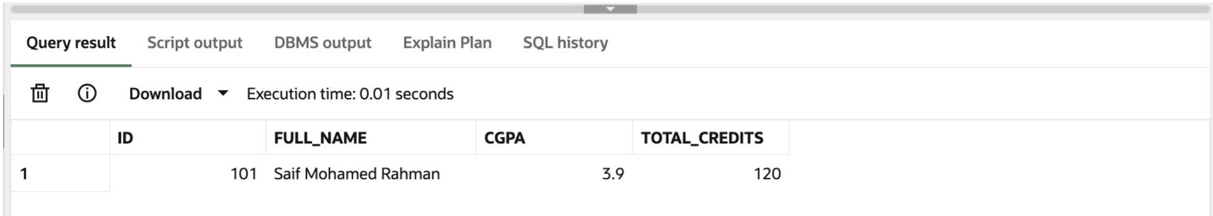
	DEGREE	INSTITUTION	GRAD_YEAR
1	SSC	Viqarunnisa Noon Sc	2016
2	HSC	Notre Dame College	2018
3	SSC	Scholars School	2017
4	HSC	Dhaka City College	2019
5	SSC	Daffodil School	2015
6	HSC	St. Joseph College	2017
7	SSC	Ideal School and Coll	2017
8	HSC	Notre Dame College	2019

Figure 3: Educational Qualification Table

1.4 Task-4: Write queries (2/3) on the data

Query 1: Write a SQL query to retrieve the ID, full name (constructed from first, middle, and last names), CGPA, and total completed credits of all students who are enrolled in the 'Computer Science and Engineering' department, have a CGPA greater than or equal to 3.5, and have completed more than 100 total credits.

```
SELECT
   s.ID,
   s.Student_Name.FirstName || ’ ’ || NVL(s.Student_Name.MiddleName, ’’) ||
       ’ ’ || s.Student_Name.LastName AS Full_Name,
   s.CGPA,
   s.Total_Credits
FROM
   Students s
WHERE
   s.Department = ’Computer Science and Engineering’
   AND s.CGPA >= 3.5
   AND s.Total_Credits > 100;
```

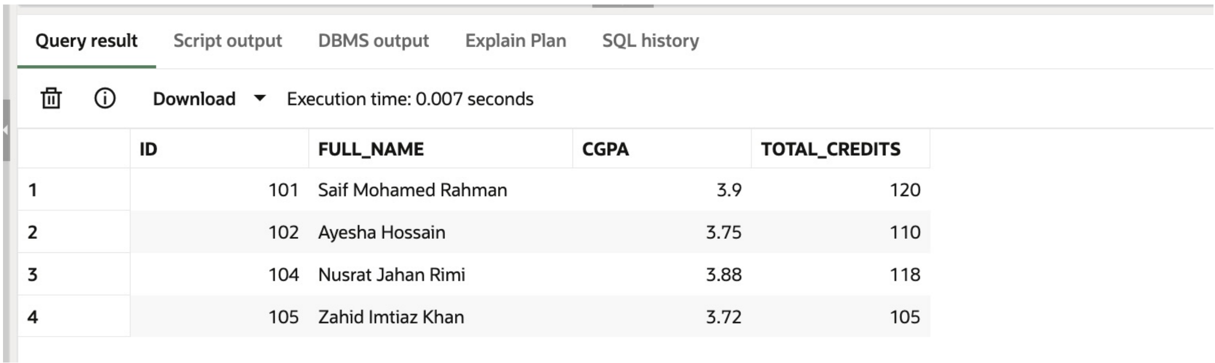


Execution time: 0.01 seconds

	ID	FULL_NAME	CGPA	TOTAL_CREDITS
1	101	Saif Mohamed Rahman	3.9	120

Query 2: Write a SQL query to find all students who have knowledge of the programming language ’Python’. The query should return each student’s ID, full name (concatenated from first, middle, and last names), CGPA, and total completed credits. The query must make use of unnesting the Prog_Knowledge varray to check for the presence of ’Python’ as a known programming language.

```
SELECT
   s.ID,
   s.Student_Name.FirstName || ’ ’ || NVL(s.Student_Name.MiddleName, ’’) ||
       ’ ’ || s.Student_Name.LastName AS Full_Name,
   s.CGPA,
   s.Total_Credits
FROM
   Students s
WHERE
   ’Python’ IN (SELECT COLUMN_VALUE FROM TABLE(s.Prog_Knowledge));
```



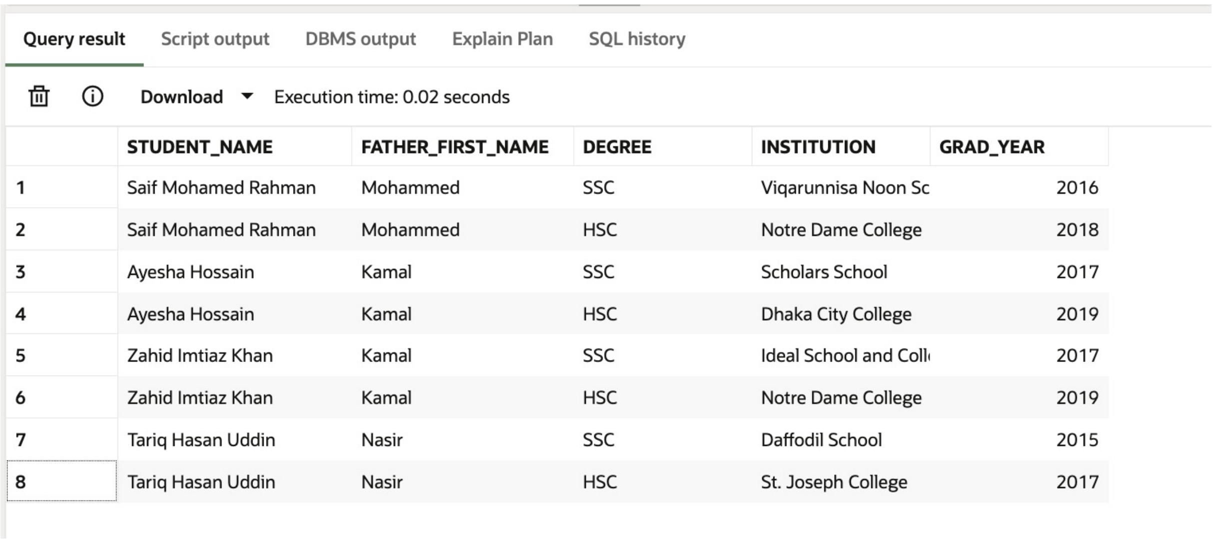
Execution time: 0.007 seconds

	ID	FULL_NAME	CGPA	TOTAL_CREDITS
1	101	Saif Mohamed Rahman	3.9	120
2	102	Ayesha Hossain	3.75	110
3	104	Nusrat Jahan Rimi	3.88	118
4	105	Zahid Imtiaz Khan	3.72	105

Query 3: Write a SQL query to retrieve the full name of each student, their father’s first

name (if available), and the details of each educational qualification they have obtained. The query should use both dereferencing of a REF type attribute (for the father's name) and unnesting of a nested table (Educations). Display the student's full name, father's first name, degree title, institution name, and graduation year.

```
SELECT
   s.Student_Name.FirstName || ’ ’ || NVL(s.Student_Name.MiddleName, ’’) ||
       ’ ’ || s.Student_Name.LastName AS Student_Name,
   f.Family_Name.FirstName AS Father_First_Name,
   edu.Degree,
   edu.Institution,
   edu.Grad_Year
FROM
   Students s
LEFT JOIN
   FamilyTrees f ON s.Father = REF(f),
   TABLE(s.Educations) edu;
```



Execution time: 0.02 seconds

	STUDENT_NAME	FATHER_FIRST_NAME	DEGREE	INSTITUTION	GRAD_YEAR
1	Saif Mohamed Rahman	Mohammed	SSC	Viqarunnisa Noon Sc	2016
2	Saif Mohamed Rahman	Mohammed	HSC	Notre Dame College	2018
3	Ayesha Hossain	Kamal	SSC	Scholars School	2017
4	Ayesha Hossain	Kamal	HSC	Dhaka City College	2019
5	Zahid Imtiaz Khan	Kamal	SSC	Ideal School and Coll	2017
6	Zahid Imtiaz Khan	Kamal	HSC	Notre Dame College	2019
7	Tariq Hasan Uddin	Nasir	SSC	Daffodil School	2015
8	Tariq Hasan Uddin	Nasir	HSC	St. Joseph College	2017

2 Part-2: NoSQL DBMS

2.1 Task-1

Using MongoDB NoSQL DBMS, define collection for students and create 5 documents to store the data of 5 students and show some queries on the documents.

2.1.1 Define collection for students

```
--
use university;
db.createCollection("students");
```

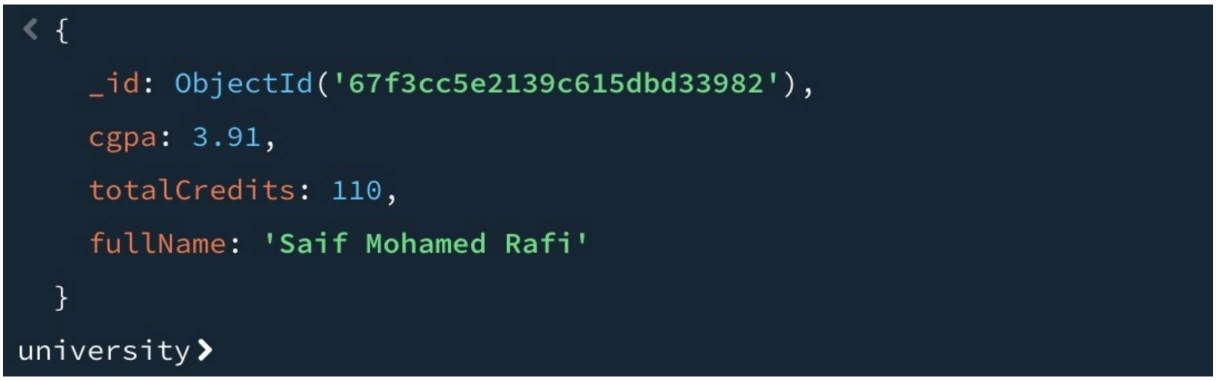
2.1.2 Insert 5 Students Data

```
db.students.insertMany([
  {
    "_id": ObjectId(),
    "studentName": { "firstName": "Saif", "middleName": "Mohamed", "lastName
        ": "Rafi" },
    "dob": ISODate("2001-03-12T00:00:00Z"),
    "cgpa": 3.91,
    "totalCredits": 110,
    "department": "Computer Science and Engineering",
    "addresses": {
      "present_address": { "street": "Banasree Block-B", "city": "Dhaka", "
          postalCode": "1219", "country": "Bangladesh" },
      "permanent_address": { "street": "Brahmanbaria Sadar", "city": "
          Brahmanbaria", "postalCode": "3400", "country": "Bangladesh" }
    },
    "phoneNumbers": ["01770589276"],
    "emailAddresses": ["saif.mohammed@northsouth.edu"],
    "researchInterests": ["AI", "RAG", "LLM", "Bioinformatics"],
    "progKnowledge": ["Python", "JavaScript", "C", "Go"],
    "educations": [{ "degree": "HSC", "institution": "Notre Dame College", "
        year": 2020 }],
    "father": { "firstName": "Mohamed", "middleName": "Zahir", "lastName": "
        Rafi" },
    "fatherOfFather": { "firstName": "Zahir", "middleName": "Ahmed", "
        lastName": "Rafi" },
    "mother": { "firstName": "Farzana", "middleName": "Nahar", "lastName": "
        Rafi" },
    "motherOfMother": { "firstName": "Jahanara", "middleName": "", "lastName
        ": "Begum" }, // Updated field
    "player": [{ "game": "Cricket", "score": 20 }],
    "organizer": [{ "club_name": "NSU ACM", "start_date": ISODate
        ("2023-01-10T00:00:00Z"), "end_date": ISODate("2024-12-31T00:00:00Z")
         }]
  },
  {
    "_id": ObjectId(),
    "studentName": { "firstName": "Zara", "middleName": "Amin", "lastName":
        "Haque" },
    "dob": ISODate("2002-08-15T00:00:00Z"),
    "cgpa": 3.67,
    "totalCredits": 98,
    "department": "Electrical and Electronic Engineering",
    "addresses": {
      "present_address": { "street": "Uttara Sector-11", "city": "Dhaka", "
          postalCode": "1230", "country": "Bangladesh" },
      "permanent_address": { "street": "Gulshan-2", "city": "Dhaka", "
          postalCode": "1212", "country": "Bangladesh" }
    },
    "phoneNumbers": ["01819233455"],
    "emailAddresses": ["zara.haque@northsouth.edu"],
    "researchInterests": ["Embedded Systems", "Signal Processing"],
    "progKnowledge": ["C", "C++", "Python"],
    "educations": [{ "degree": "HSC", "institution": "Viqarunnisa Noon
        School and College", "year": 2020 }],
    "father": { "firstName": "Sami", "middleName": "Mahbub", "lastName": "
        Haque" },
    "fatherOfFather": { "firstName": "Mahbub", "middleName": "Kamal", "
        lastName": "Haque" },
    "mother": { "firstName": "Lubna", "middleName": "Sultana", "lastName": "
        Chowdhury" },
    "motherOfMother": { "firstName": "Rokeya", "middleName": "", "lastName":
         "Begum" }, // Updated field
    "player": [{ "game": "Badminton", "score": 8 }],
    "organizer": [{ "club_name": "IEEE NSU", "start_date": ISODate
        ("2024-01-01T00:00:00Z"), "end_date": ISODate("2024-12-01T00:00:00Z")
         }]
  },
  {
    "_id": ObjectId(),
    "studentName": { "firstName": "Tanvir", "middleName": "Ahmed", "lastName
        ": "Nashit" },
    "dob": ISODate("2000-01-27T00:00:00Z"),
    "cgpa": 3.45,
    "totalCredits": 102,
    "department": "Computer Science and Engineering",
    "addresses": {
      "present_address": { "street": "Baridhara DOHS", "city": "Dhaka", "
          postalCode": "1206", "country": "Bangladesh" },
      "permanent_address": { "street": "Mirpur-1", "city": "Dhaka", "
          postalCode": "1216", "country": "Bangladesh" }
    },
    "phoneNumbers": ["01711666789"],
    "emailAddresses": ["tanvir.nashit@northsouth.edu"],
    "researchInterests": ["Data Science", "Cybersecurity"],
    "progKnowledge": ["Python", "R", "Java"],
    "educations": [{ "degree": "HSC", "institution": "Adamjee Cantonment
        College", "year": 2019 }],
    "father": { "firstName": "Anwar", "middleName": "Hossain", "lastName": "
        Nashit" },
    "fatherOfFather": { "firstName": "Rafiq", "middleName": "Ahmed", "
        lastName": "Nashit" },
    "mother": { "firstName": "Salma", "middleName": "Begum", "lastName": "
        Nashit" },
    "motherOfMother": { "firstName": "Fatema", "middleName": "", "lastName":
         "Begum" }, // Updated field
    "player": [],
    "organizer": []
  },
  {
    "_id": ObjectId(),
    "studentName": { "firstName": "Ayesha", "middleName": "Rahman", "
        lastName": "Mitu" },
    "dob": ISODate("2003-04-05T00:00:00Z"),
    "cgpa": 3.80,
    "totalCredits": 85,
    "department": "Architecture",
    "addresses": {
      "present_address": { "street": "Mohakhali DOHS", "city": "Dhaka", "
          postalCode": "1212", "country": "Bangladesh" },
      "permanent_address": { "street": "Chattogram", "city": "Chattogram", "
          postalCode": "4000", "country": "Bangladesh" }
    },
    "phoneNumbers": ["01911223344"],
    "emailAddresses": ["ayesha.mitu@northsouth.edu"],
    "researchInterests": ["Sustainable Design", "Urban Planning"],
    "progKnowledge": ["AutoCAD", "Revit", "SketchUp"],
    "educations": [{ "degree": "HSC", "institution": "Chattogram Govt. Girls
         College", "year": 2021 }],
    "father": { "firstName": "Shams", "middleName": "Uddin", "lastName": "
        Rahman" },
    "fatherOfFather": { "firstName": "Habib", "middleName": "Uddin", "
        lastName": "Rahman" },
    "mother": { "firstName": "Nazmun", "middleName": "Nahar", "lastName": "
        Mitu" },
    "motherOfMother": { "firstName": "Shahana", "middleName": "", "lastName
        ": "Begum" }, // Updated field
    "player": [],
    "organizer": []
  },
  {
    "_id": ObjectId(),
    "studentName": { "firstName": "Imran", "middleName": "Khan", "lastName":
         "Nabil" },
    "dob": ISODate("2002-12-09T00:00:00Z"),
    "cgpa": 3.25,
    "totalCredits": 75,
    "department": "BBA",
    "addresses": {
      "present_address": { "street": "Khilgaon", "city": "Dhaka", "postalCode
          ": "1219", "country": "Bangladesh" },
      "permanent_address": { "street": "Sylhet Sadar", "city": "Sylhet", "
          postalCode": "3100", "country": "Bangladesh" }
    },
    "phoneNumbers": ["01600112233"],
    "emailAddresses": ["imran.nabil@northsouth.edu"],
    "researchInterests": ["Marketing Analytics", "Entrepreneurship"],
    "progKnowledge": ["Excel", "SPSS", "Tableau"],
    "educations": [{ "degree": "HSC", "institution": "Sylhet Govt College",
        "year": 2020 }],
    "father": { "firstName": "Rashid", "middleName": "Ahmed", "lastName": "
        Khan" },
    "fatherOfFather": { "firstName": "Rahmat", "middleName": "Ali", "
        lastName": "Khan" },
    "mother": { "firstName": "Nusrat", "middleName": "Jahan", "lastName": "
        Nabil" },
    "motherOfMother": { "firstName": "Jamila", "middleName": "", "lastName":
         "Begum" }, // Updated field
    "player": [{ "game": "Football", "score": 6 }],
    "organizer": [{ "club_name": "NSU YES!", "start_date": ISODate
        ("2024-05-01T00:00:00Z"), "end_date": ISODate("2025-01-15T00:00:00Z")
         }]
  }
]);
```

2.1.3 Sample Queries

Query 1: Write a SQL query to retrieve the ID, full name (constructed from first, middle, and last names), CGPA, and total completed credits of all students who are enrolled in the 'Computer Science and Engineering' department, have a CGPA greater than or equal to 3.5, and have completed more than 100 total credits.

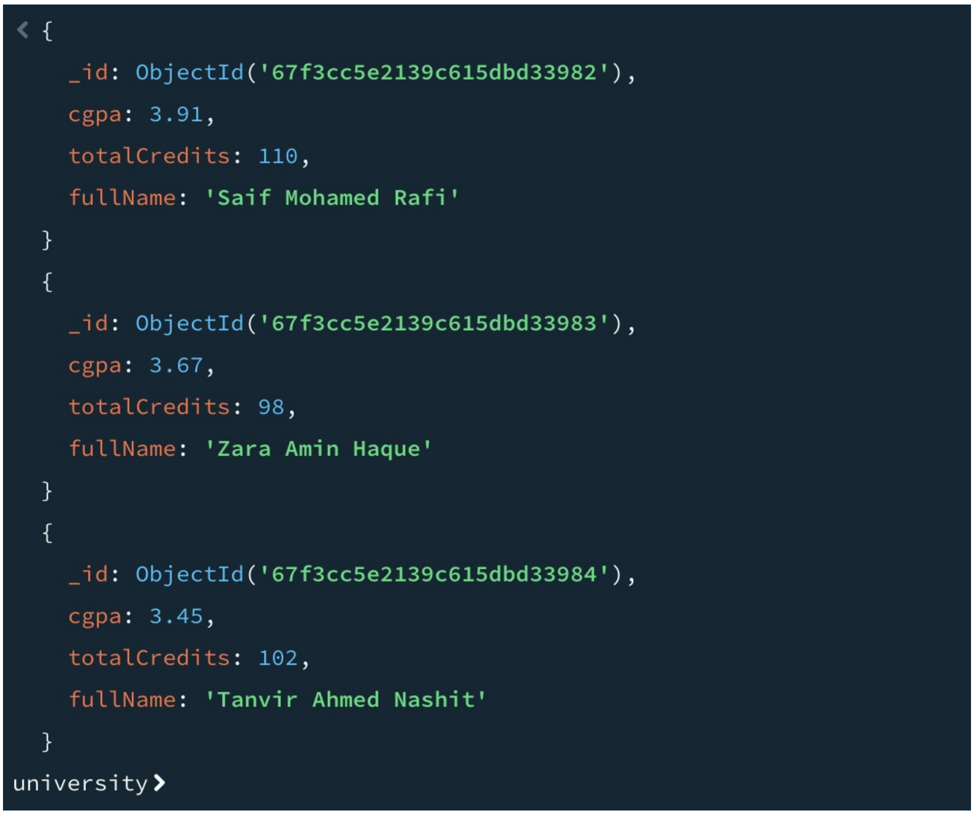
```
db.students.aggregate([
  {
    $match: {
      "department": "Computer Science and Engineering",
      "cgpa": { $gte: 3.5 },
      "totalCredits": { $gt: 100 }
    }
  },
  {
    $project: {
      _id: 1,
      fullName: {
        $concat: [
          "$studentName.firstName",
          " ",
          "$studentName.middleName",
          " ",
          "$studentName.lastName"
        ]
      },
      cgpa: 1,
      totalCredits: 1
    }
  }
])
```



```
< {
    _id: ObjectId('67f3cc5e2139c615dbd33982'),
    cgpa: 3.91,
    totalCredits: 110,
    fullName: 'Saif Mohamed Rafi'
  }
university>
```

Query 2: Write a SQL query to find all students who have knowledge of the programming language 'Python'. The query should return each student's ID, full name (concatenated from first, middle, and last names), CGPA, and total completed credits. The query must make use of unnesting the Prog_Knowledge varray to check for the presence of 'Python' as a known programming language.

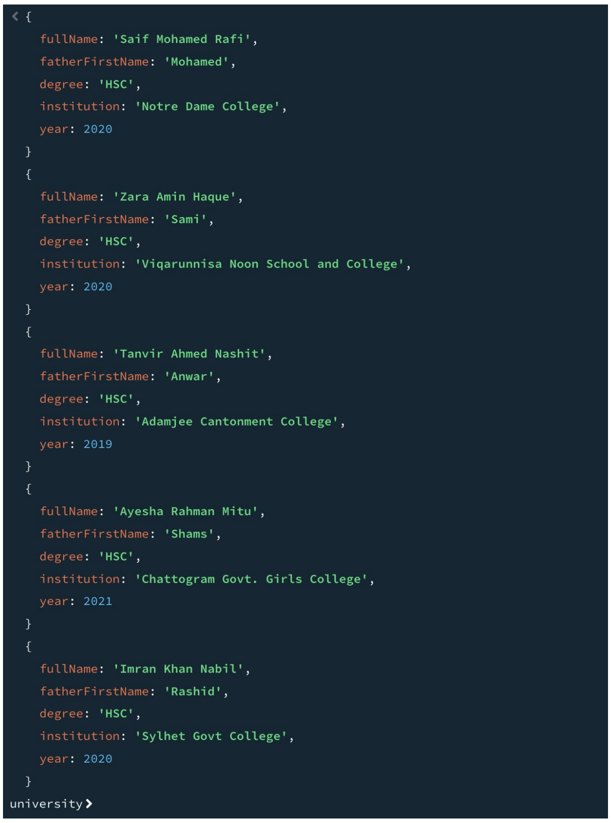
```
db.students.aggregate([
  {
    $match: {
      "progKnowledge": "Python"
    }
  },
  {
    $project: {
      _id: 1,
      fullName: {
        $concat: [
          "$studentName.firstName",
          " ",
          "$studentName.middleName",
          " ",
          "$studentName.lastName"
        ]
      },
      cgpa: 1,
      totalCredits: 1
    }
  }
]);
```



```
< {
    _id: ObjectId('67f3cc5e2139c615dbd33982'),
    cgpa: 3.91,
    totalCredits: 110,
    fullName: 'Saif Mohamed Rafi'
  }
  {
    _id: ObjectId('67f3cc5e2139c615dbd33983'),
    cgpa: 3.67,
    totalCredits: 98,
    fullName: 'Zara Amin Haque'
  }
  {
    _id: ObjectId('67f3cc5e2139c615dbd33984'),
    cgpa: 3.45,
    totalCredits: 102,
    fullName: 'Tanvir Ahmed Nashit'
  }
university>
```

Query 3: Write a SQL query to retrieve the full name of each student, their father's first name (if available), and the details of each educational qualification they have obtained. The query should use both dereferencing of a REF type attribute (for the father's name) and unnesting of a nested table (Educations). Display the student's full name, father's first name, degree title, institution name, and graduation year.

```
db.students.aggregate([
  {
    $unwind: "$educations"
  },
  {
    $project: {
      _id: 0,
      fullName: {
        $concat: [
          "$studentName.firstName",
          " ",
          "$studentName.middleName",
          " ",
          "$studentName.lastName"
        ]
      },
      fatherFirstName: "$father.firstName",
      degree: "$educations.degree",
      institution: "$educations.institution",
      year: "$educations.year"
    }
  }
]);
```



```
< {
    fullName: 'Saif Mohamed Rafi',
    fatherFirstName: 'Mohamed',
    degree: 'HSC',
    institution: 'Notre Dame College',
    year: 2020
  }
  {
    fullName: 'Zara Amin Haque',
    fatherFirstName: 'Sami',
    degree: 'HSC',
    institution: 'Viqarunnisa Noon School and College',
    year: 2020
  }
  {
    fullName: 'Tanvir Ahmed Nashit',
    fatherFirstName: 'Anwar',
    degree: 'HSC',
    institution: 'Adamjee Cantonment College',
    year: 2019
  }
  {
    fullName: 'Ayesha Rahman Mitu',
    fatherFirstName: 'Shams',
    degree: 'HSC',
    institution: 'Chattogram Govt. Girls College',
    year: 2021
  }
  {
    fullName: 'Imran Khan Nabil',
    fatherFirstName: 'Rashid',
    degree: 'HSC',
    institution: 'Sylhet Govt College',
    year: 2020
  }
university>
```

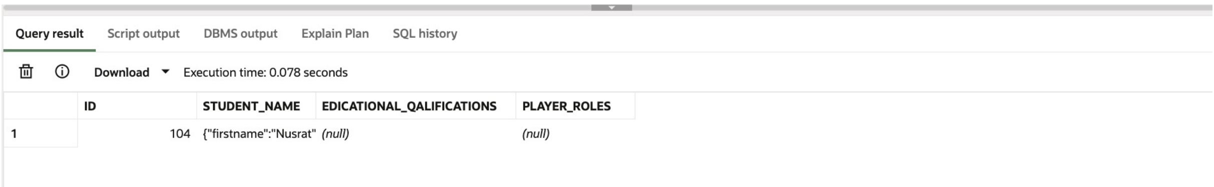
2.2 Task-2

For optional fields, data become sparse because of many null values in Relational or Object-Relational model. But in NoSQL model, it does not happen. Describe this from your implementation.

2.2.1 ORDBMS Implementation (Oracle)

```
-- insert a new student into students table without "
   Educational_Qualification", "Player" (Optional Fields)

INSERT INTO Students VALUES (
104,
Name_Type(Nusrat, Jahan, Rimi),
DATE 2001-11-22,
3.88,
118,
Architecture,
Address_Type(
Address_Detail_Type(H-44, New Eskaton, Ramna, Dhaka),
Address_Detail_Type(H-21, Road-13, Mohakhali, Dhaka)
),
Phone_Number_List(01744444444),
Email_Address_List(nusrat.jahan@northsouth.edu, nusrat.jr@gmail.com)
,
Research_Interest_List(Urban Planning, Sustainable Design, 3D
Modeling),
Programming_Language_List(AutoCAD, SketchUp, Python),
NULL,
(SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 1),
(SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 2),
(SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 3),
(SELECT REF(f) FROM FamilyTrees f WHERE f.Family_ID = 4),
NULL,
Organizer_Roles_List(Organizer_Type(NSU Earth Club, DATE 2022-01-15,
DATE 2023-01-15))
);
```

```
-- Oracle DBMS query

Select ID, STUDENT_NAME, EDUCATIONS AS Edicational_Qalifications,
   PLAYER_ROLES
From Students
where ID=104;
```



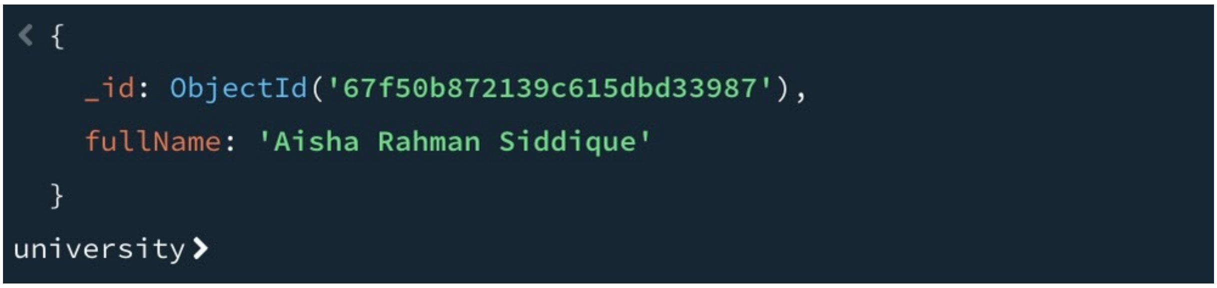
	ID	STUDENT_NAME	EDICATIONAL_QALIFICATIONS	PLAYER_ROLES
1	104	{"firstname":"Nusrat"	(null)	(null)

Figure 4: Oracle ORDBMS Sparse Data (NULL)

2.2.2 NoSQL Implementation (MongoDB)

```
-- insert a new student into students table without "researchInterests", "
   progKnowledge" (Optional Fields)

db.students.insertOne({
  _id: ObjectId(),
  studentName: { firstName: "Aisha", middleName: "Rahman", lastName: "
      Siddique" },
  dob: ISODate("2001-05-15T00:00:00Z"),
  cgpa: 3.60,
  totalCredits: 88,
  department: "Electrical Engineering",
  addresses: {
    present_address: { street: "Bashundhara R/A", city: "Dhaka", postalCode:
        "1229", country: "Bangladesh" },
    permanent_address: { street: "Chittagong City", city: "Chittagong",
        postalCode: "4000", country: "Bangladesh" }
  },
  phoneNumbers: ["01877665544"],
  emailAddresses: ["aisha.siddique@northsouth.edu"],
  educations: [{ degree: "HSC", institution: "Viqarunnisa Noon College",
      year: 2019 }],
  father: { firstName: "Faruq", middleName: "Ahmed", lastName: "Siddique" },
  fatherOfFather: { firstName: "Karim", middleName: "Box", lastName: "
      Siddique" },
  mother: { firstName: "Samina", middleName: "Yasmin", lastName: "Rahman" },
  motherOfMother: { firstName: "Rokeya", middleName: "", lastName: "Begum"
      },
  player: [
      { game: "Table Tennis", score: 15 }
  ],
  organizer: [
      { club_name: "NSU Photography Club", start_date: ISODate("2023-09-01T00
          :00:00Z"), end_date: ISODate("2024-08-31T00:00:00Z") }
  ]
})
```

```
-- MongoDB Query

db.students.aggregate([
  {
    $match: {
      "studentName.firstName": "Aisha"
    }
  },
  {
    $project: {
      _id: 1,
      fullName: {
        $concat: [
          "$studentName.firstName",
          " ",
          "$studentName.middleName",
          " ",
          "$studentName.lastName"
        ]
      },
      researchInterests: 1,
      progKnowledge: 1
    }
  }
])
```



```
< {
    _id: ObjectId('67f50b872139c615dbd33987'),
    fullName: 'Aisha Rahman Siddique'
  }
university>
```

Figure 5: NoSQL Sparse Data for NULL values (MongoDB)

In the Oracle Object-Relational implementation, the ‘Students‘ table has a predefined, fixed structure. Even though ‘EDUCATIONS‘ and ‘PLAYER_ROLES‘ are optional, columns must exist for them within the table definition. Consequently, when inserting the record for student ID 104, who lacks educational qualifications and player roles in this specific entry, explicit ‘NULL‘ values are inserted into those respective columns. The database allocates storage space for these ‘NULL‘ markers, representing the absence of data. As shown in the Oracle query result for ID 104, these columns appear with ‘NULL‘ values, illustrating how storage is consumed even for missing optional data, leading to potential data sparseness if many students have missing information in these fields.

Conversely, the NoSQL implementation using MongoDB demonstrates a flexible schema approach. When inserting the student document for ”Aisha”, the optional fields ‘researchInterests‘ and ‘progKnowledge‘ were simply omitted entirely from the document structure because no data was provided for them. Unlike Oracle, MongoDB does not require predefined placeholders for every possible field. It only stores the fields and values that are actually present in a given document. The subsequent MongoDB query result for ”Aisha” confirms this; the ‘researchInterests‘ and ‘progKnowledge‘ fields are completely absent from the output document, not shown as ‘NULL‘ or empty. This dynamic nature inherently avoids the data sparseness problem associated with many optional fields, as no storage is wasted on representing missing data.