American International University-Bangladesh (AIUB)

Faculity Of Science & Technology



Course Title

INTRODUCTION TO DATABASE

Spring 2023-2024

Section: I

Project Name : **Birth Record Management System**

Supervised By

Juena Ahmed Noshin

Submitted By:

|  |  |
| --- | --- |
| Name | ID |
| MD. ASICKUZZAMAN | 23-50985-1 |
| ISRAT JAHAN CHOWDHURY | 23-51219-1 |
| JAFIR ISLAM SIAM | 23-50574-1 |
| MD SADMAN SAKIB SHAD | 22-49501-3 |

**Contents: Page Number**

1. Introduction 1
2. Scenario Description 2
3. ER Diagram 3
4. Normalization 4-7
5. Finalization 8
6. Shema Diagram 9
7. Table Creation 10-17
8. Data Insertion 18-20
9. Query Writing 21-25

10.Relation Algebra 26

11.Conclusion 27

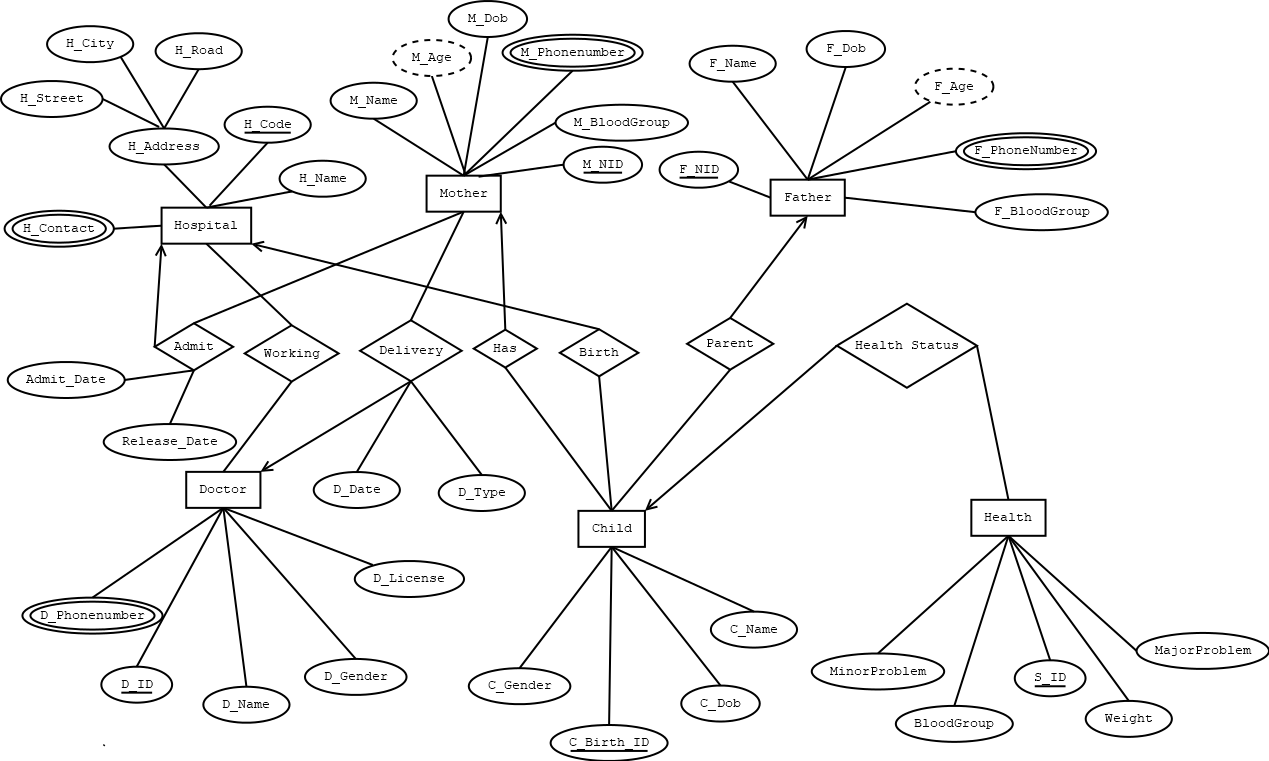
**1.Introduction**

The efficient administration of birth records is crucial for many institutions and organizations, including medical facilities, governmental bodies, and legal entities, in the fast-paced world of today. A birth record management system is a centralized platform that is intended to record, preserve, and handle important data pertaining to births that take place inside a certain jurisdiction or healthcare system. It is essential to maintaining the reliability, accessibility, and integrity of birth-related data because it makes important administrative, medical, and legal procedures easier. Simplifying the tracking and documentation of births from the time of delivery till formal birth certificates are issued is the main goal of a birth record management system. Such a system seeks to replace labor-intensive, paper-based record-keeping procedures with effective, digital workflows by utilizing contemporary database technologies and information management techniques. This change not only makes data more accurate and comprehensive, but it also makes the birth registration procedure more responsive and efficient overall.

1. **Scenario Description:**

In a Birth Management System a child can be born in only one hospital whereas in a hospital many child’s can be born. A child is identified by Birth Id. The system also stores the Child name, date of birth and gender . A child can have one father and mother whereas they can have multiple children. Father and Mother are indentified by their NID respectively. Both of them has name, date of birth, contact number and blood group. A mother can be admitted into one hospital and in one hospital many mothers can be admitted. The system stores the admit and release date of mothers. A doctor delivers the child of the mother and the delivery of many mothers can be done by one doctor. The type of the delivery and delivery date is also stored in the system. A doctor is identified by the Doctor ID. License, gender, contact number and name of the Doctor is also stored in the system. A doctor can work in only one hospital and many doctors can work in a single hospital. A Hospital is identified by Hospital Code. The system also stores the name, address and contact of the hospital. After the birth, Health Status of each child is stored in the system after checkup. Health status of a child is identified by Status ID. The system stores the weight, blood group, major or minor diseases of the child if have any.

**3. ER DIAGRAM:**

**f**

**NORMALIZATION**

**Parent :**

|  |
| --- |
| UNF  Parent(F\_NID,F\_Name,F\_BloodGroup ,F\_PhoneNumber,C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob) |
| 1NF:  F\_PhoneNumber is the mutlivalued attribute  1)F\_NID,F\_Name,F\_BloodGroup ,F\_PhoneNumber,C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob |
| 2NF :  1)F\_NID,F\_Name,F\_BloodGroup ,F\_PhoneNumber  2)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob |
| 3NF:  There is no transitive dependancy.Relation already in 3NF  1)F\_NID,F\_Name,F\_BloodGroup ,F\_PhoneNumber  2)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob |

**Has :**

|  |
| --- |
| UNF  Has ( C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob,M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber) |
| 1NF:  C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob,M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber |
| 2NF :  1)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob  2)M\_NID,M\_Name,M\_BloodGroup M\_PhoneNumber |
| 3NF:  There is no transitive dependency. Relation already 3NF  1)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob  2)M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber |

**Birth :**

|  |
| --- |
| UNF  Birth(H\_Name,H\_Code,H\_Contact,H\_Address ,H\_ Street, H\_City,H\_Road,C\_Name  C\_Birth\_ID,C\_Gender,C\_Dob) |
| 1NF:  H\_Contact is the multivalued attribute  1)H\_Name,H\_Code,H\_Contact,H\_Address , H\_ Street, H\_City,H\_Road,C\_Name ,C\_Birth\_ID,C\_Gender,C\_Dob |
| 2NF:  1)H\_Name,H\_Code,H\_Contact,H\_Address ,H\_ Street, H\_City,H\_Road  2)C\_Name ,C\_Birth\_ID,C\_Gender,C\_Dob |
| 3NF:  1)H\_Name,H\_Code,H\_Contact 2)H\_Address H\_ Street, H\_City,H\_Road 3)C\_Name,C\_Birth\_ID,C\_Gender,C\_Dob |

**Admit:**

|  |
| --- |
| UNF  (M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber,H\_Name,H\_Code,H\_Contact,H\_Address   H\_ Street, H\_City,H\_Road,Admit\_Date,Release\_Date) |
| 1NF:  H\_Contact and M\_PhoneNumber is the multivalued attribute    1)M\_NID,M\_Name,M\_BloodGroup  M\_PhoneNumber,H\_Name,H\_Code,H\_Contact,H\_Address   H\_ Street, H\_City,H\_Road |
| 2NF:  1)M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber  2)H\_Name,H\_Code,H\_Contact,H\_Address   H\_ Street, H\_City,H\_Road |
| 3NF:  1)M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber  2)H\_Name,H\_Code,H\_Contact  3)H\_Address  H\_ Street, H\_City,H\_Road |

**Delivery:**

|  |
| --- |
| UNF  (M\_NID,M\_Name,M\_BloodGroup  M\_PhoneNumber,D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License,Delivery\_Date,  Delivery\_Type) |
| 1NF:  M\_PhoneNumber and D\_PhoneNumber is the multivalued attribute  1)M\_NID,M\_Name,M\_BloodGroup  M\_PhoneNumber,D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |
| 2NF:  1)M\_NID,M\_Name,M\_BloodGroup  M\_PhoneNumber  2)D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |
| 3NF:  There is no transitive dependency.Relation is already 3NF    1)M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber  2)D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |

**Working**

|  |
| --- |
| UNF  (H\_Name,H\_Code,H\_Contact,H\_Address ,H\_ Street, H\_City,H\_Road,D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License) |
| 1NF:  H\_Contact and D\_PhoneNumber is the multivalued attribute    1)H\_Name,H\_Code,H\_Contact,H\_Address   H\_ Street, H\_City,H\_Road,D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |
| 2NF:  1)H\_Name,H\_Code,H\_Contact,H\_Address   H\_ Street, H\_City,H\_Road  2)D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |
| 3NF:  1)H\_Name,H\_Code,H\_Contact  2)H\_Address ,H\_ Street, H\_City,H\_Road  3)D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License |

**Health Status:**

|  |
| --- |
| UNF  (C\_Name,C\_Birth\_ID,C\_Gender,C\_Dob,S\_ID,BloodGroup,Weight,MajorProblem, MinorProblem) |
| 1NF:  There is no multivalued attribute.Relation is already 1NF  1)C\_Name C\_Birth\_ID,C\_Gender,C\_Dob,S\_ID,BloodGroup,Weight,MajorProblem, MinorProblem |
| 2NF:  1)C\_Name C\_Birth\_ID,C\_Gender,C\_Dob  2)S\_ID,BloodGroup,Weight,MajorProblem,MinorProblem |
| 3NF:  There is no transitive dependency.Relation is already 3NF.  1)C\_Name  C\_Birth\_ID,C\_Gender,C\_Dob  2)S\_ID,BloodGroup,Weight,MajorProblem,MinorProblem |

**Temporary Tables:**

1)F\_NID,F\_Name,,F\_BloodGroup,F\_PhoneNumber

2)~~C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob~~,**F\_NID.**

3~~)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob~~,**M\_NID.**

1. ~~M\_NID,M\_Name,M\_BloodGroup,M\_PhoneNumber~~
2. ~~H\_Name,H\_Code,H\_Contact,~~**~~HA\_ID~~**
3. ~~HA\_ID,H\_Address H\_ Street, H\_City,H\_Road~~
4. ~~C\_Name,C\_Birth\_ID,C\_Gender,C\_Dob,~~**H\_Code**
5. ~~M\_NID,M\_Name,M\_BloodGroup,M\_PhoneNumbe~~r,Admit\_Date,Release\_Date,

**H\_Code**

1. ~~H\_Name,H\_Code,H\_Contact,~~**~~HA\_ID~~**
2. HA\_ID,H\_Address, H\_ Street, H\_City,H\_Road
3. M\_NID,M\_Name,M\_BloodGroup,M\_PhoneNumber,Delivery\_Date, Delivery\_Type,**D\_ID**
4. ~~D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License~~
5. H\_Name,H\_Code,H\_Contact,**HA\_ID**
6. HA\_ID,H\_Address,H\_ Street, H\_City,H\_Road
7. D\_ID,D\_Name,D\_PhoneNumber,D\_License,**H\_Code**
8. C\_Name,C\_Birth\_ID,C\_Gender,C\_Dob,**S\_ID**

17)S\_ID,BloodGroup,Weight,MajorProblem,MinorProblem

**Final Tables:**

1)F\_NID,F\_Name,F\_BloodGroup,F\_PhoneNumber1,F\_PhoneNumber2

2)C\_Name,C\_Birth\_ID, C\_Gender,C\_Dob,F\_NID,M\_NID, H\_Code, S\_ID

3)HA\_ID,H\_Address H\_ Street, H\_City,H\_Road

4))M\_NID,M\_Name,M\_BloodGroup ,M\_PhoneNumber1,M\_PhoneNumber2,Admit\_Date,

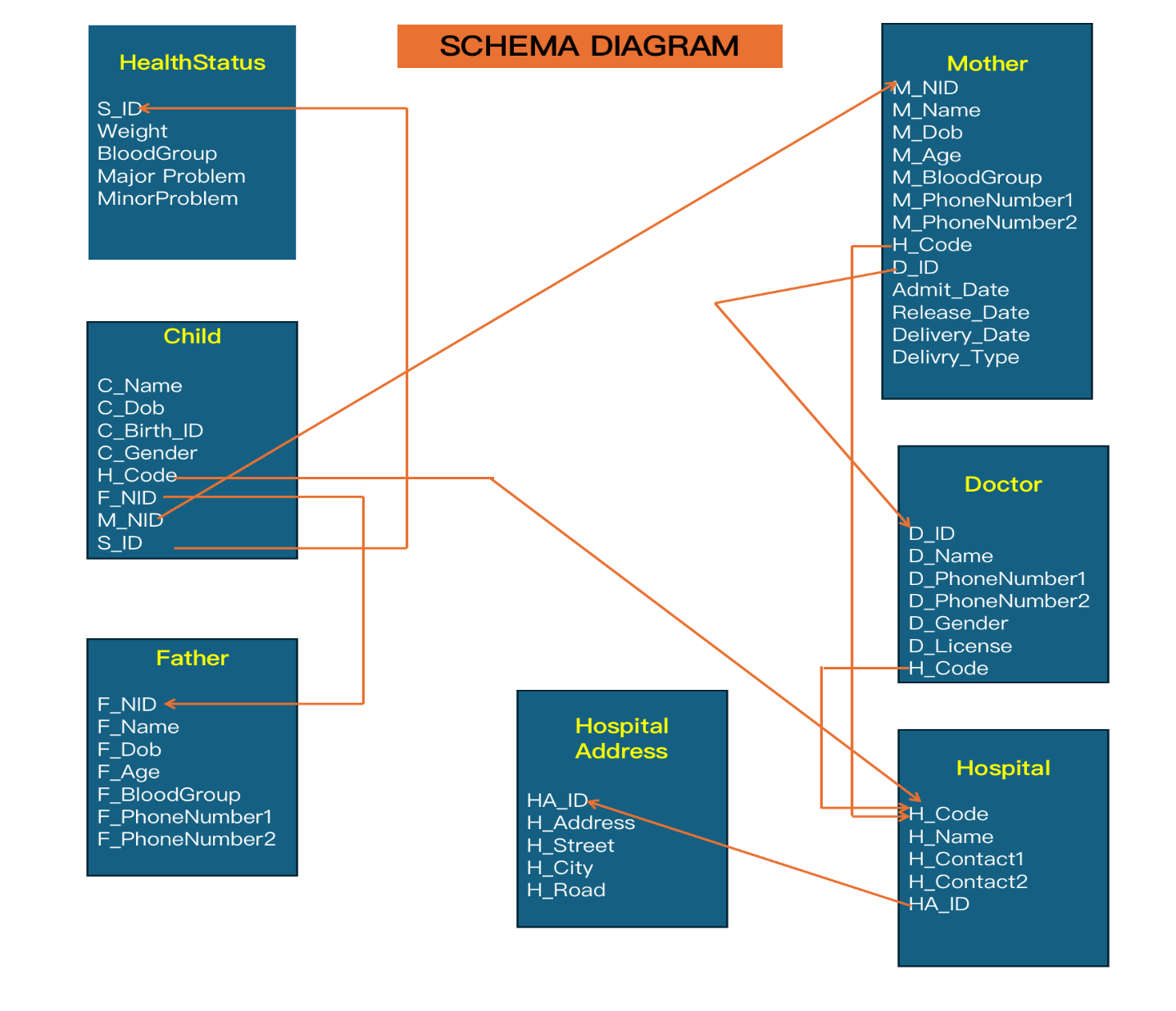
Release\_Date,Delivery\_Date,Delivery\_Type,H\_Code, D\_ID

5)H\_Name,H\_Code,H\_Contact,HA\_ID

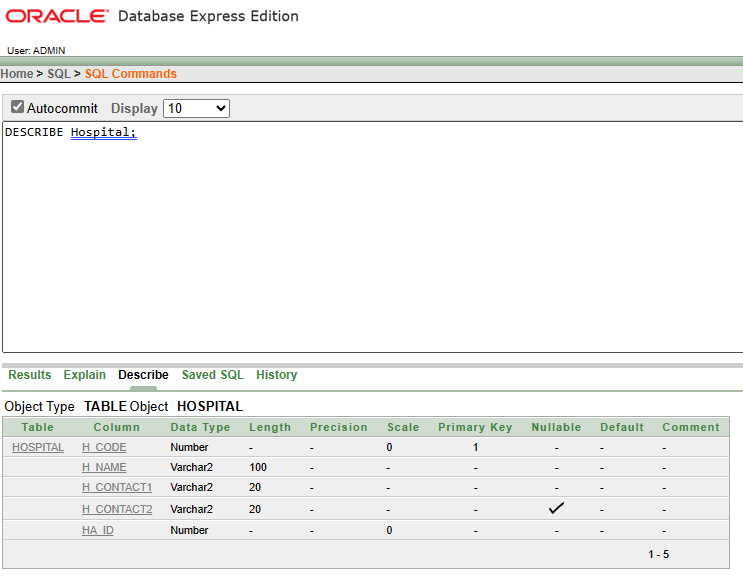
6)D\_ID,D\_Name,D\_Gender,D\_PhoneNumber,D\_License,H\_Code

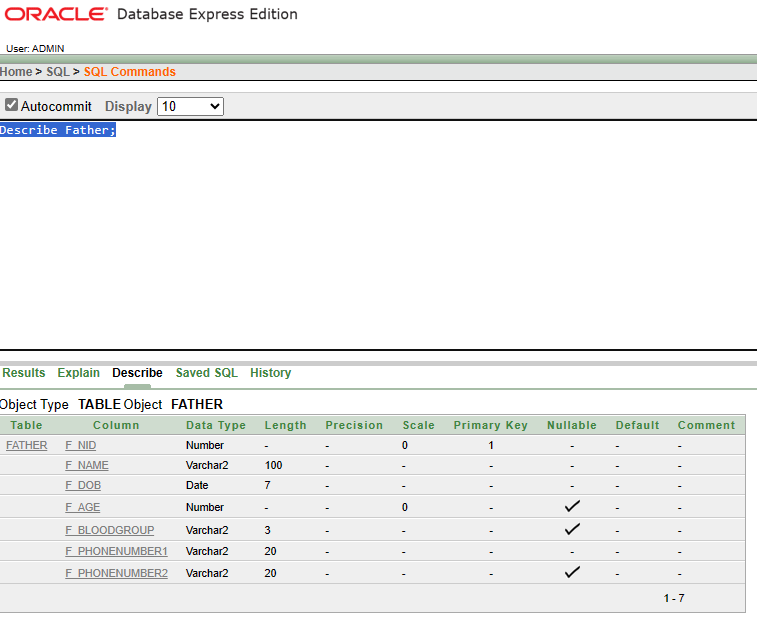
7)S\_ID,BloodGroup,Weight,MajorProblem,MinorProblem

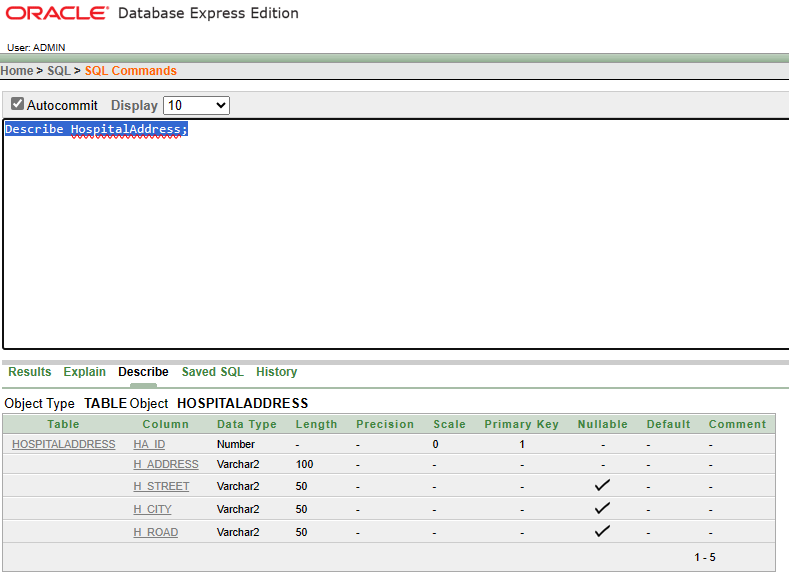
**SCHEMA DIAGRAM**

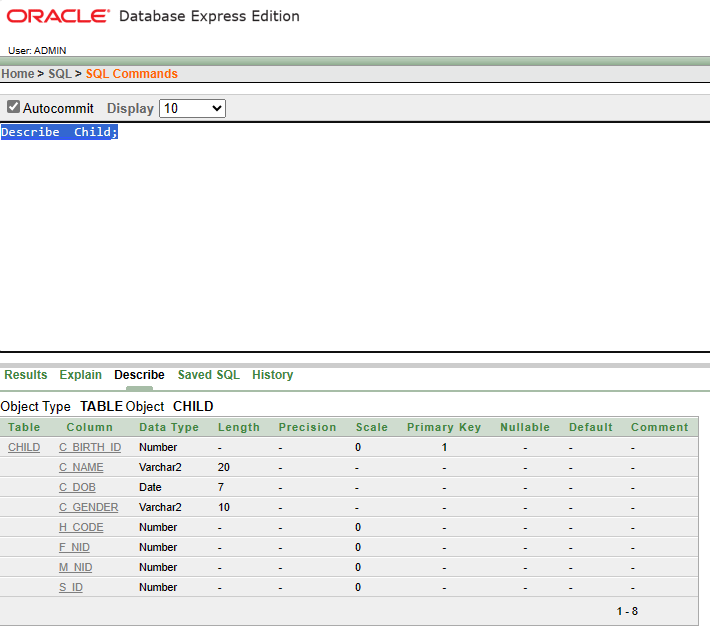
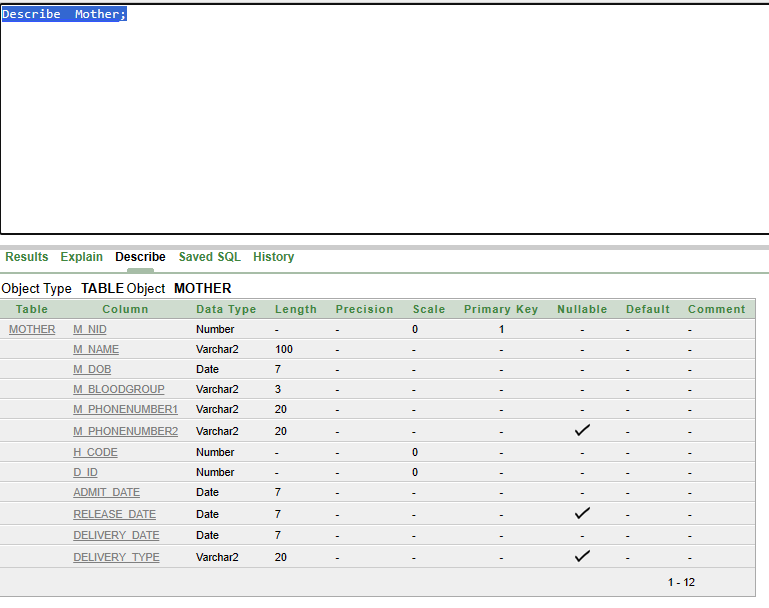
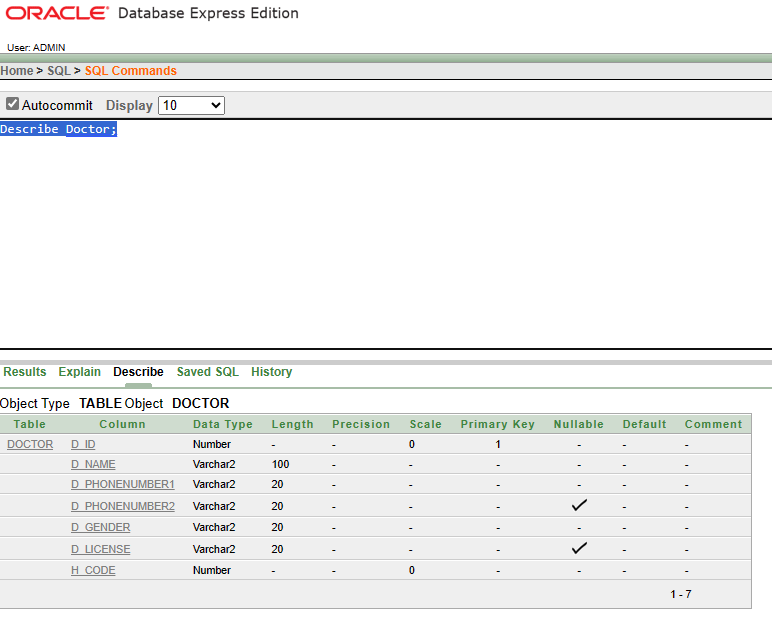


**TABLE CREATION**







**Table creation queries:**

Create Table HealthStatus (  
    S\_ID INT PRIMARY KEY,   
    Weight NUMBER(10,2) NOT NULL CHECK (Weight > 0),  
    BloodGroup VARCHAR2(3) CHECK (BloodGroup IN ('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-')),  
    MajorProblem VARCHAR2(100),  
    MinorProblem VARCHAR2(100)  
);

Create Table Father(  
    F\_NID int primary key,   
F\_Name varchar2 (100) not null,  
F\_Dob Date not null,  
F\_Age int,  
F\_BloodGroup varchar2(3) Check (F\_BloodGroup In('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-')),  
F\_PhoneNumber1 varchar2(20) not null,  
F\_PhoneNumber2 varchar2(20)  
);

Create Table HospitalAddress(  
    HA\_ID int primary key,   
H\_Address varchar2(100) not null,  
H\_Street varchar2(50),  
H\_City varchar2(50),  
H\_Road varchar2(50) );

Create Table Hospital(  
    H\_Code int primary key,   
H\_Name varchar2(100) not null,  
H\_Contact1 varchar2(20) not null,  
H\_Contact2 varchar2(20) ,  
HA\_ID int not null,  
    FOREIGN KEY(HA\_ID) REFERENCES HospitalAddress(HA\_ID)  
);

Create Table Doctor(  
    D\_ID int primary key,  
D\_Name varchar2(100) not null,  
D\_PhoneNumber1 varchar2(20) not null,  
D\_PhoneNumber2 varchar2(20),  
D\_Gender  varchar2(20) not null Check (D\_Gender In ('Male', 'Female')),  
D\_License varchar2(20) unique ,  
H\_Code int not null,  
    Foreign key (H\_Code) References Hospital (H\_Code)  
);

Create Table Mother(  
M\_NID int primary key,  
M\_Name varchar2(100) not null,  
M\_Dob Date not null,  
M\_BloodGroup varchar2(3) not null,  
M\_PhoneNumber1  varchar2(20) not null,  
M\_PhoneNumber2 varchar2(20) ,  
H\_Code int not null,  
D\_ID int not null,  
Admit\_Date Date not null,  
Release\_Date Date,  
Delivery\_Date Date not null,  
Delivery\_Time Time(0),  
Delivery\_Type  boolean check (Delivery\_Type In(0,1)),  
    Foreign key (H\_Code) References Hospital(H\_Code),  
    Foreign key (D\_ID) References Doctor(D\_ID)  
);

Create Table Child  (  
    C\_Birth\_ID int primary key,  
    C\_Name varchar2 (20) not null,  
C\_Dob Date not null,  
C\_Gender varchar2(10) not null Check (C\_Gender In ('Male', 'Female')),  
H\_Code int not null,  
F\_NID int not null,  
M\_NID int not null,  
S\_ID int not null,  
    Foreign key (H\_Code) References Hospital(H\_Code),  
    Foreign key (F\_NID) References Father(F\_NID),  
    Foreign key (M\_NID) References Mother(M\_NID),  
    Foreign key (S\_ID ) References HealthStatus(S\_ID)  
    );

**Sequence queries:**

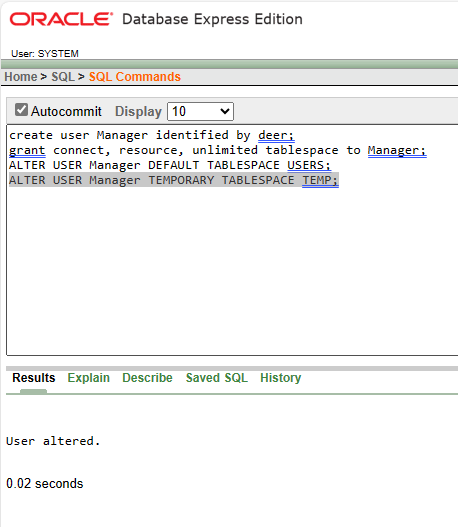
Create Sequence Doctor\_Key\_Sequence  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;

Create Sequence Hospital\_Key\_Sequence  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;

Create Sequence Health\_Key\_Sequence  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;

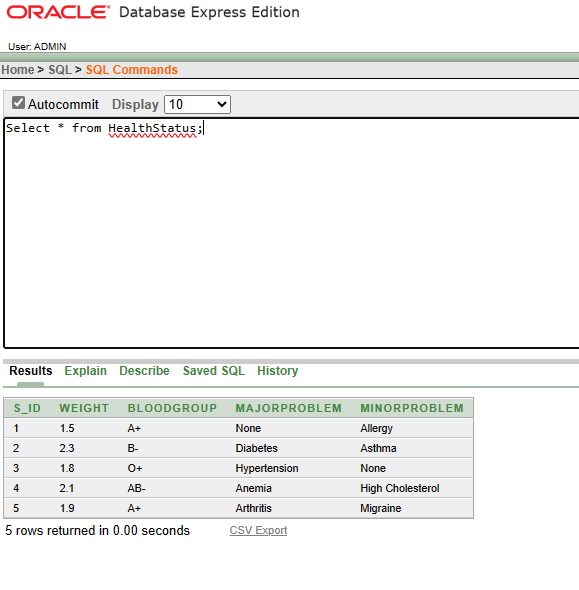
Create Sequence HAddress\_Key\_Sequence  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;

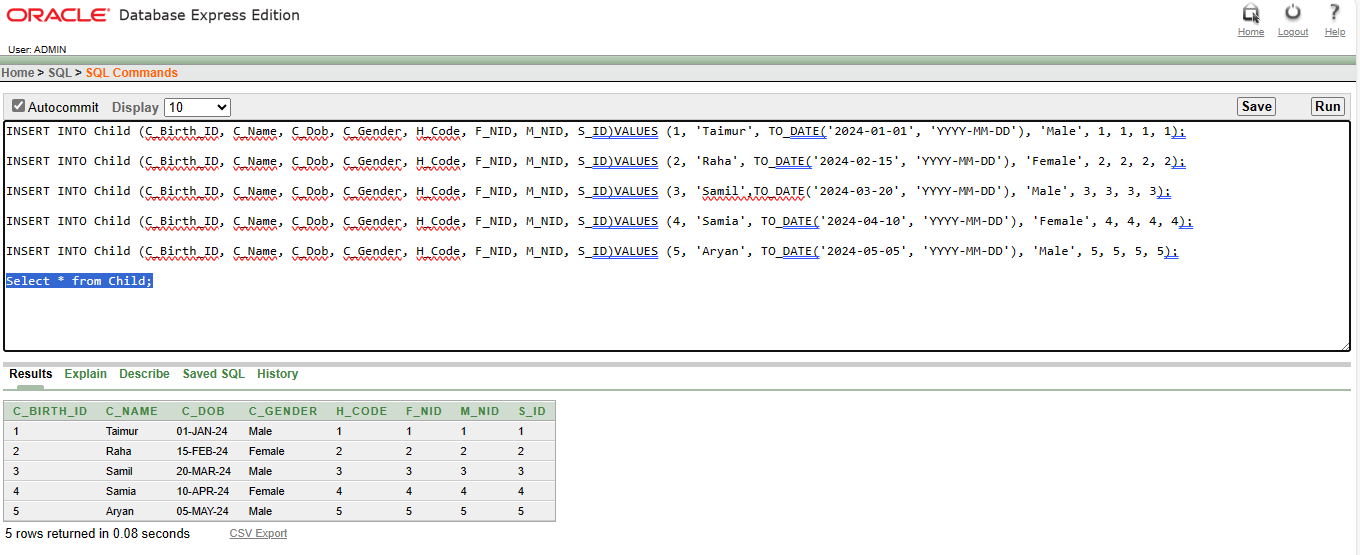
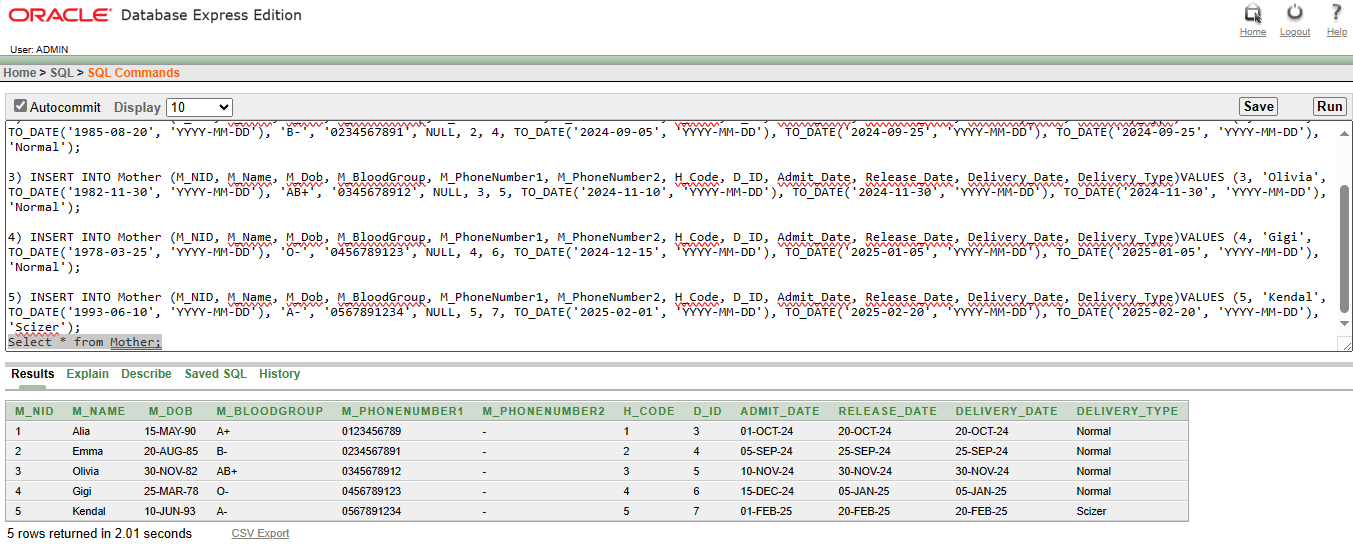
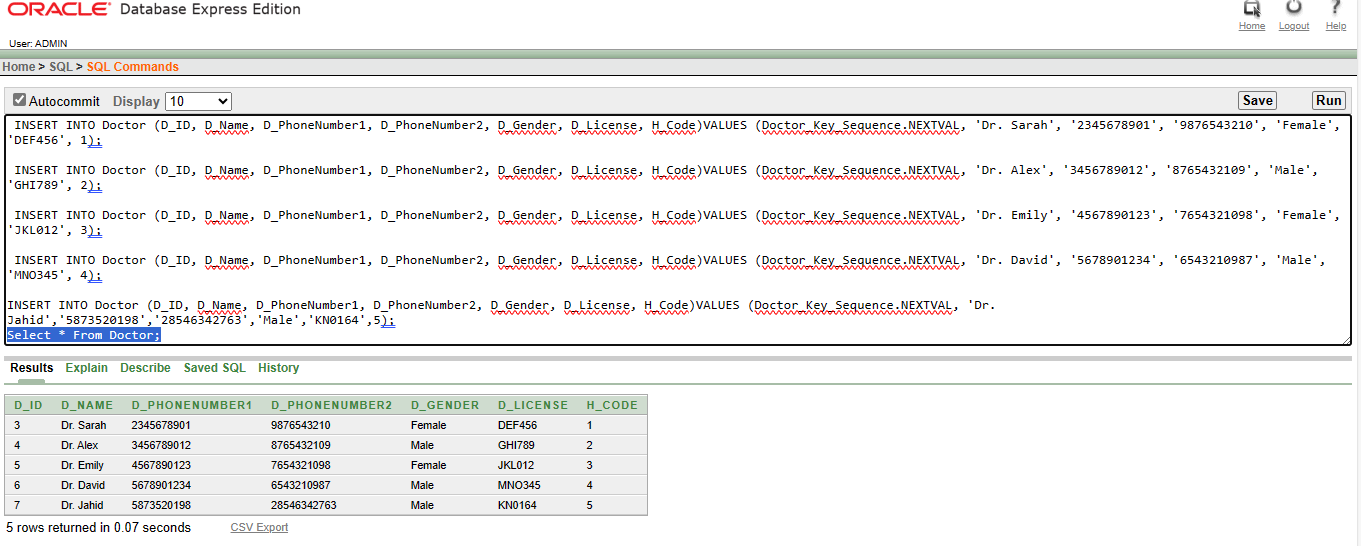
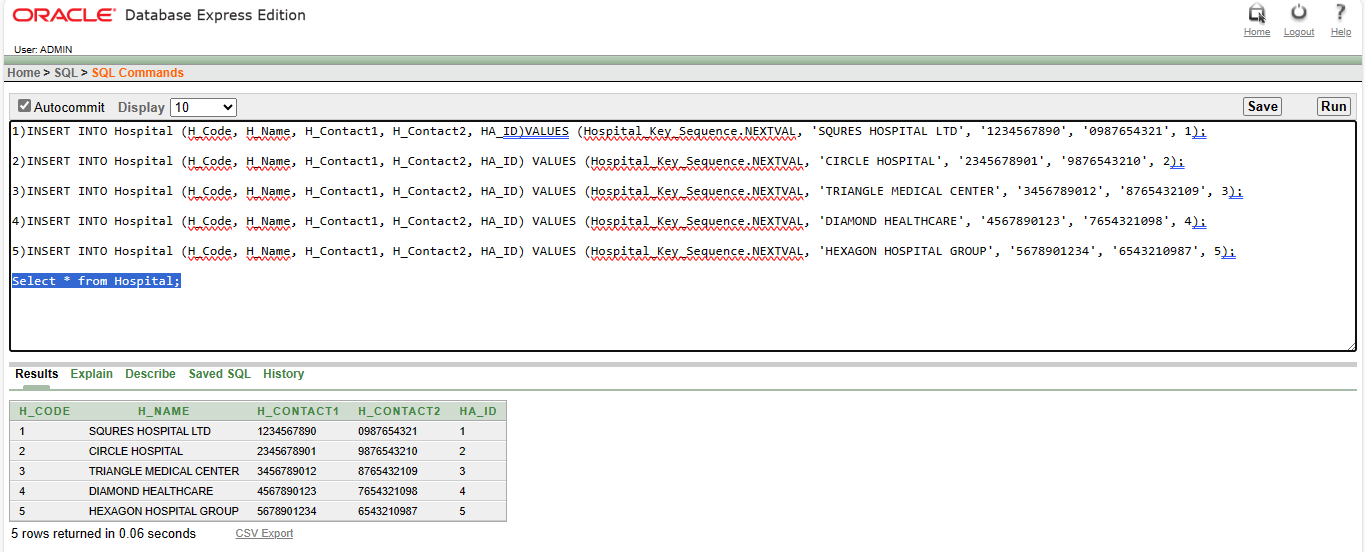
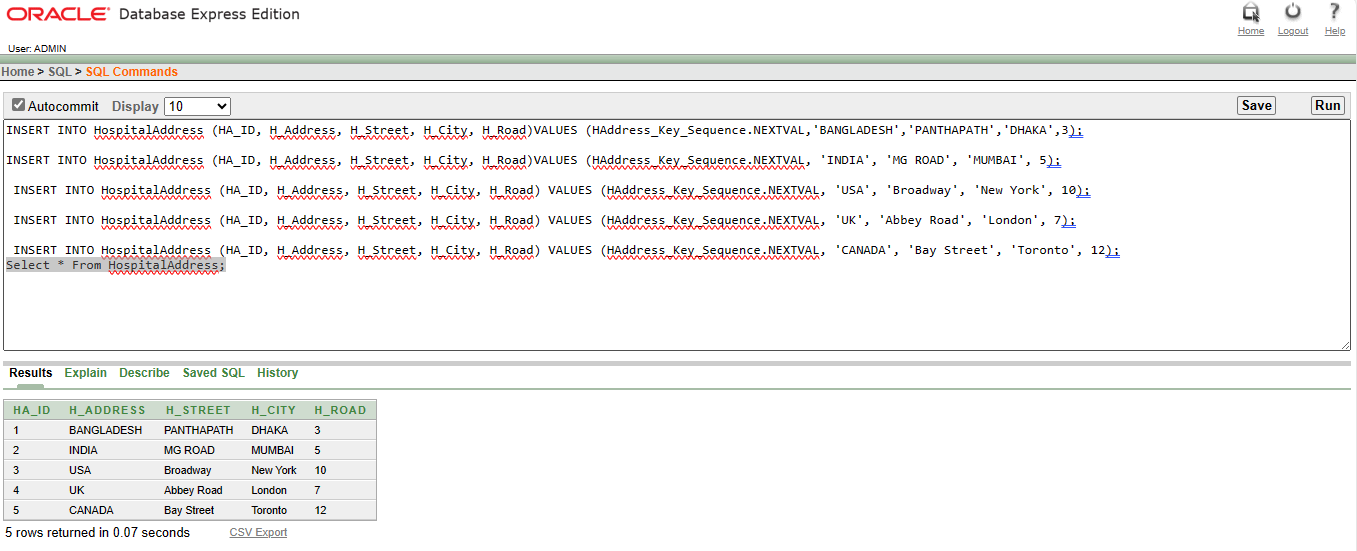
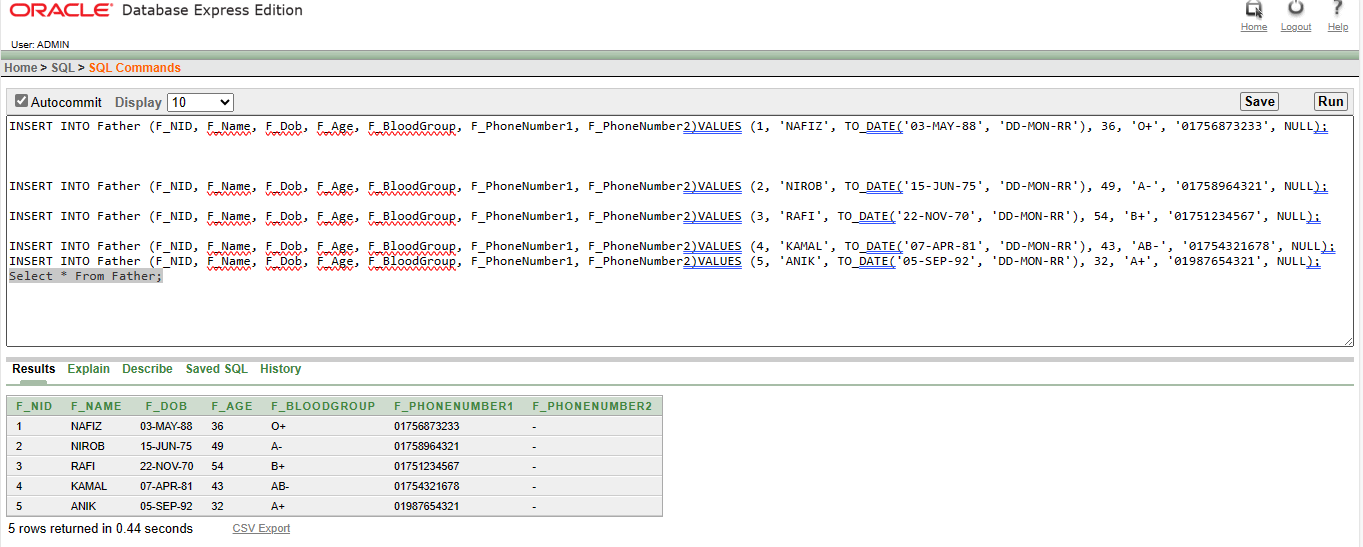
New User:



**Data Insertion**

INSERT INTO HealthStatus (S\_ID, Weight, BloodGroup, MajorProblem, MinorProblem)VALUES (1, 1.5, 'A+', 'None', 'Allergy');





**Query Writing:**

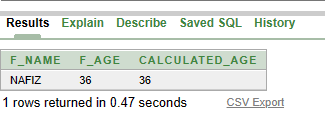
Query 1: Single Row Function

Question: Retrieve the age of the father named 'NAFIZ' from the 'Father' table.

SELECT F\_Name, F\_Age,

TRUNC(MONTHS\_BETWEEN(SYSDATE, F\_Dob) / 12) AS Calculated\_Age

FROM Father WHERE F\_Name = 'NAFIZ';



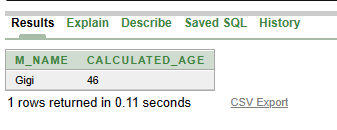
Query 2: Single Row Function

Question: Retrieve the age of the mother named 'Gigi' from the 'Mother' table.

SELECT M\_Name,

EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM M\_Dob) AS Calculated\_Age

FROM Mother WHERE M\_Name = 'Gigi';

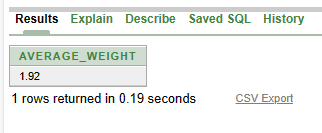


Query 3: Group Function

Question: Calculate the average weight of all children.

SELECT AVG(Weight) AS Average\_Weight

FROM HealthStatus;



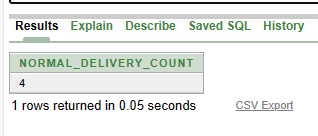
Query 4: Group Function

Question: Count the number of mothers admitted for normal delivery.

SELECT COUNT(\*) AS Normal\_Delivery\_Count

FROM Mother

WHERE Delivery\_Type = 'Normal';



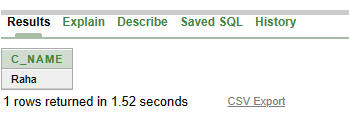
Query 5: Subquery

Question: Retrieve the names of the children born in hospitals located in India.

SELECT C\_Name

FROM Child

WHERE H\_Code IN (SELECT HA\_ID FROM HospitalAddress WHERE H\_Address = 'INDIA');



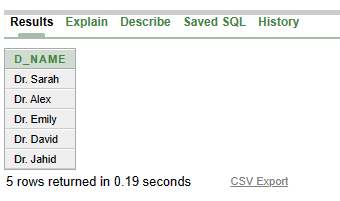
Query 6: Subquery

Question: Retrieve the names of the doctors working at hospitals where children with major health problems are born.

SELECT D\_Name

FROM Doctor

WHERE H\_Code IN (SELECT DISTINCT H\_Code FROM Child WHERE S\_ID IN (SELECT S\_ID FROM HealthStatus WHERE MajorProblem IS NOT NULL));



Query 7: Joining

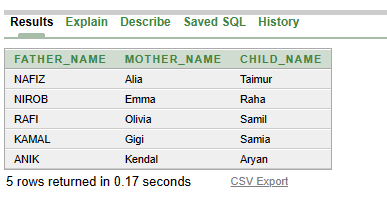
Question: Retrieve the names of fathers and mothers along with their children's names.

SELECT F.F\_Name AS Father\_Name, M.M\_Name AS Mother\_Name, C.C\_Name AS Child\_Name

FROM Father F

JOIN Child C ON F.F\_NID = C.F\_NID

JOIN Mother M ON C.M\_NID = M.M\_NID;



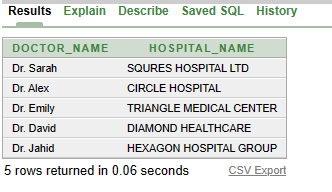
Query 8: Joining

Question: Retrieve the names of doctors along with the hospitals they work in.

SELECT D.D\_Name AS Doctor\_Name, H.H\_Name AS Hospital\_Name

FROM Doctor D

JOIN Hospital H ON D.H\_Code = H.H\_Code;



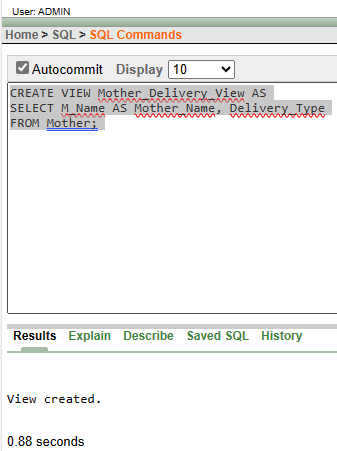
Query 9: View

Question: Create a view to display the names of mothers and their delivery types.

CREATE VIEW Mother\_Delivery\_View AS

SELECT M\_Name AS Mother\_Name, Delivery\_Type

FROM Mother;



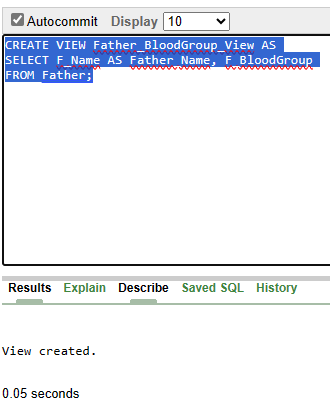
Query 10: View

Question: Create a view to display the names of fathers and their blood groups.

CREATE VIEW Father\_BloodGroup\_View AS

SELECT F\_Name AS Father\_Name, F\_BloodGroup

FROM Father;



**Relational Algebra**

**Quesitons:**

1)Find the name of the child whose birth id is 4 ?

2)Find the name of the hospital whose hospital code is 5?

3)Find the hospital id whose hospital city is MUMBAI?

4)Find the delivery date where the mother name is Alia ?

5)Find the name of the mother where blood group is B+?

**Answers:**

1. ∏ C\_Name(σC\_Birth\_ID=1(Child))
2. ∏H\_Name(σH\_Code=5(Hospital))
3. ∏HA\_ID(σH\_City=”MUMBAI”(HospitalAddress))
4. ∏Delivery\_Date (σM\_Name=”Alia”(Delivery))
5. ∏M\_Name(σM\_BloodGroup=”B+”(Mother))

**Conclusion**

To sum up, this project Birth Management System provides a strong foundation for effectively managing birth-related data in healthcare settings. The system maintains data integrity and makes it easier to retrieve and modify birth records by grouping data into distinct tables and creating relationships between entities. Future development will concentrate on creating an intuitive user interface, improving system functionality, integrating with other systems, putting data analytics skills into practice, and perhaps creating a mobile application. These improvements are intended to better the delivery of healthcare, optimize workflows, and offer insightful information to both patients and healthcare practitioners.