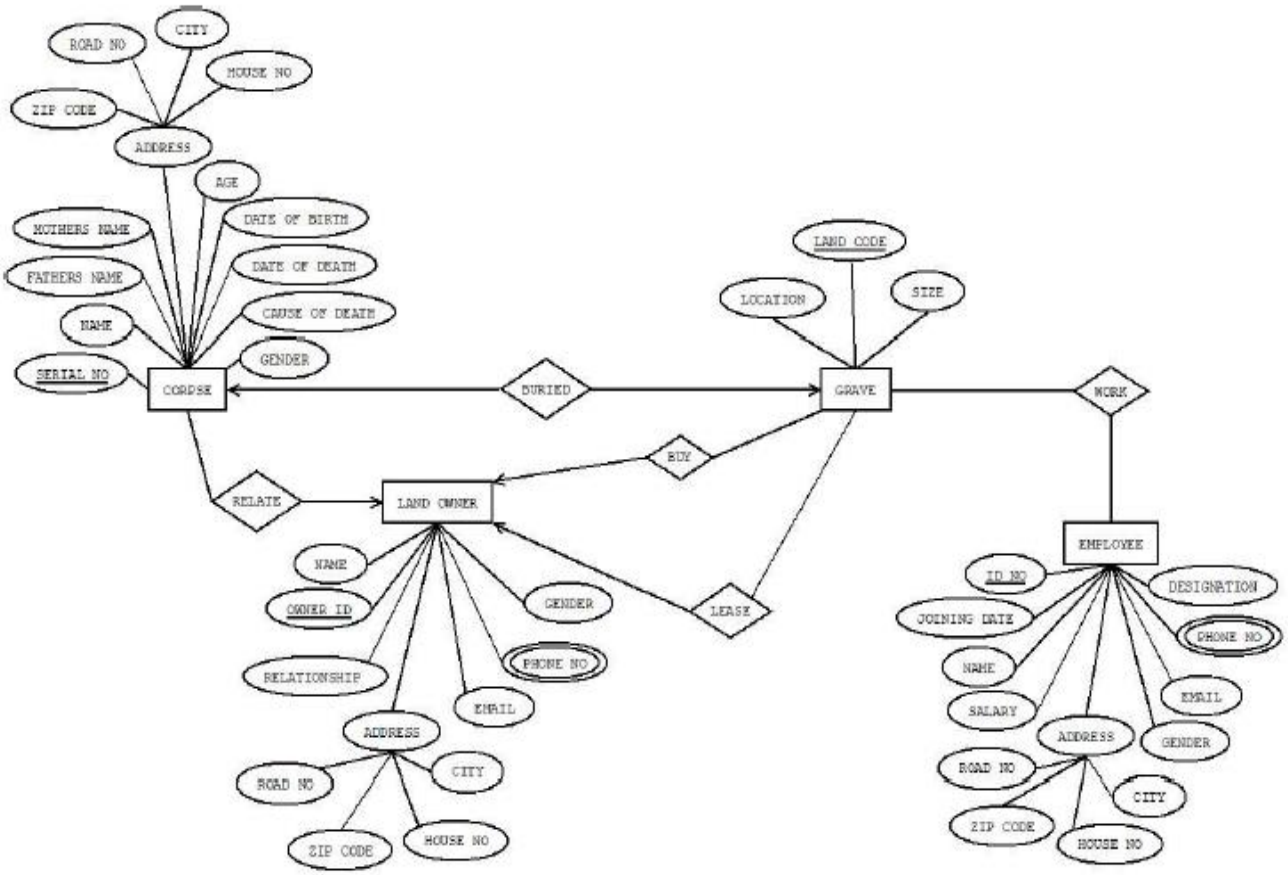
Name	ID	STUDENT SIGN
Md Tuhin Reza	18-39074-3	

Class Test 04

1. Normalize the ER Diagram given below up to 3rd Normal Form and finalize the tables that needs to be created. Then (in Oracle using SQL) write down the queries that are required to create all the tables with necessary constraints. Also insert at least 3 rows of data in each created table.



Answer Box (Normalization steps in detail as shown in Normalization Tutorial Slide + all the queries required to create the tables and insert data after Normalization):

Corpse:

Attributes: serial no (primary key), Name, Father Name, Mother name, Age, Date of birth, Date of Death, Corpse of Date, Gender, zip code, road no, city, house no.

Normalization:

1NF:

The address is non-atomic as it consists of multiple attributes: zip code, road no, city, house no. These need to be split.

2NF:

Since serial no is the primary key and all other columns directly depend on it, we're already satisfying 2NF.

3NF:

The address attributes (zip code, road no, city, house no) are more related to the location than the corpse itself. These can be abstracted into a separate table to satisfy 3NF.

Tables:

Corpse:

Attributes: serial no (primary key), Name, Father Name, Mother name, Age, Date of birth, Date of Death, Corpse of Date, Gender, AddressID (foreign key).

Address:

Attributes: AddressID (primary key), zip code, road no, city, house no.

Land Owner:

Attributes: Owner Id (primary key), Name, Relationship, Address (zip code, road no, city, house no), Gender, Phone Number, Email.

Normalization:

1NF:

The Address is non-atomic, consisting of multiple attributes: zip code, road no, city, house no. These should be split.

The Phone Number attribute can have multiple values, which is also non-atomic and needs to be split into a separate table.

2NF:

Since Owner Id is the primary key and all other columns directly depend on it, we're already in 2NF.

3NF:

Address attributes (zip code, road no, city, house no) are related more to a location than the Land Owner itself. These can be abstracted into a separate table.

Each owner can have multiple phone numbers. This relationship should also be abstracted into its own table to eliminate transitive dependencies.

Tables:

Land Owner:

Attributes: Owner Id (primary key), Name, Relationship, AddressID (foreign key), Gender, Email.

Address:

Attributes: AddressID (primary key), zip code, road no, city, house no.

PhoneNumbers:

Attributes: PhoneNumberID (primary key), OwnerID (foreign key), PhoneNumber.

Employee:

Attributes: id no (primary key), joining date, Name, salary, Address (zip code, road no, city, house no), gender, email, phone number, designation.

Normalization:

1NF:

The Address is non-atomic, consisting of multiple attributes: zip code, road no, city, house no. These should be split.

The Phone Number attribute can have multiple values, which is non-atomic and needs to be split into a separate table.

2NF:

Since id no is the primary key and all other columns directly depend on it, we're already in 2NF.

3NF:

Address attributes (zip code, road no, city, house no) are related more to a location than the Employee himself/herself. These can be abstracted into a separate table.

Each employee can have multiple phone numbers. This relationship should also be abstracted into its own table to eliminate transitive dependencies.

Tables:

Employee:

Attributes: id no (primary key), joining date, Name, salary, AddressID (foreign key), gender, email, designation.

EmployeeAddress:

Attributes: AddressID (primary key), zip code, road no, city, house no.

EmployeePhoneNumbers:

Attributes: PhoneNumberID (primary key), EmployeeID (foreign key), PhoneNumber.

Grave:

Attributes: Location, Land Code (primary key), size.

Normalization:

1NF:

All columns already contain atomic values.

All columns are unique.

The order doesn't matter.

The table appears to be in 1NF.

2NF:

Since Land Code is the primary key and there's no compound primary key, there's no possibility of partial dependency. Therefore, the table is in 2NF.

3NF:

The table does not appear to have any transitive dependencies. All non-key attributes are directly dependent on the primary key and are not dependent on any other non-key attribute.

Table:

Grave

Attributes: Location, Land Code (primary key), size.

Temporary Tables:

Corpse

Attributes: serial no (primary key), Name, Father Name, Mother name, Age, Date of birth, Date of Death, Corpse of Date, Gender, AddressID (foreign key).

Address

Attributes: AddressID (primary key), zip code, road no, city, house no.

LandOwner

Attributes: Owner Id (primary key), Name, Relationship, AddressID (foreign key), Gender, Email.

PhoneNumbersForLandOwners

Attributes: PhoneNumberID (primary key), OwnerID (foreign key), PhoneNumber.

Employee

Attributes: id no (primary key), joining date, Name, salary, AddressID (foreign key), gender, email, designation.

EmployeeAddress

Attributes: AddressID (primary key), zip code, road no, city, house no.

EmployeePhoneNumbers

Attributes: PhoneNumberID (primary key), EmployeeID (foreign key), PhoneNumber.

Grave

Attributes: Location, Land Code (primary key), size.

Final Table:

Corpse

Attributes: serial no (primary key), Name, Father Name, Mother name, Age, Date of birth, Date of Death, Corpse of Date, Gender, AddressID (foreign key).

LandOwner

Attributes: Owner Id (primary key), Name, Relationship, AddressID (foreign key), Gender, Email.

Employee

Attributes: id no (primary key), joining date, Name, salary, AddressID (foreign key), gender, email, designation.

Address

Attributes: AddressID (primary key), zip code, road no, city, house no. This table can be linked to Corpse, LandOwner, and Employee using the AddressID.

PhoneNumbers

Attributes: PhoneNumberID (primary key), EntityID (can be either OwnerID or EmployeeID), EntityType (to distinguish between LandOwner and Employee), PhoneNumber.

Grave

Attributes: Location, Land Code (primary key), size.

Address Table

```
CREATE TABLE Address (
   AddressID NUMBER PRIMARY KEY,
   zip_code VARCHAR2(10),
   road_no VARCHAR2(50),
   city VARCHAR2(50),
   house_no VARCHAR2(50)
);
INSERT INTO Address VALUES (1, '10001', 'Road 1', 'CityA', '1A');
INSERT INTO Address VALUES (2, '10002', 'Road 2', 'CityB', '2B');
INSERT INTO Address VALUES (3, '10003', 'Road 3', 'CityC', '3C');
```

Corpse Table

```
CREATE TABLE Corpse (
   serial_no NUMBER PRIMARY KEY,
   Name VARCHAR2(100),
   FatherName VARCHAR2(100),
   MotherName VARCHAR2(100),
   Age NUMBER(3),
   DateOfBirth DATE,
   DateOfDeath DATE,
   CorpseOfDate DATE,
   Gender CHAR(1),
   AddressID NUMBER,
   FOREIGN KEY (AddressID) REFERENCES Address(AddressID)
```

);

INSERT INTO Corpse VALUES (1, 'John Doe', 'FatherDoe', 'MotherDoe', 30, '1990-01-01', '2020-01-01', '2020-01-02', 'M', 1);

INSERT INTO Corpse VALUES (2, 'Jane Smith', 'FatherSmith', 'MotherSmith', 28, '1992-01-01', '2020-01-01', '2020-01-02', 'F', 2);

INSERT INTO Corpse VALUES (3, 'Robert Brown', 'FatherBrown', 'MotherBrown', 29, '1991-01-01', '2020-01-01', '2020-01-02', 'M', 3);

LandOwner Table

CREATE TABLE LandOwner (

OwnerId NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Relationship VARCHAR2(50),

Gender CHAR(1),

Email VARCHAR2(100),

AddressID NUMBER,

FOREIGN KEY (AddressID) REFERENCES Address(AddressID)

);

INSERT INTO LandOwner VALUES (1, 'OwnerA', 'Friend', 'M', 'ownerA@email.com', 1);

INSERT INTO LandOwner VALUES (2, 'OwnerB', 'Relative', 'F', 'ownerB@email.com', 2);

INSERT INTO LandOwner VALUES (3, 'OwnerC', 'Acquaintance', 'M', 'ownerC@email.com', 3);

Employee Table

CREATE TABLE Employee (

id_no NUMBER PRIMARY KEY,

```
    joining_date DATE,
    Name VARCHAR2(100),
    salary NUMBER(10,2),
    gender CHAR(1),
    email VARCHAR2(100),
    designation VARCHAR2(50),
    AddressID NUMBER,
    FOREIGN KEY (AddressID) REFERENCES Address(AddressID)
);
INSERT INTO Employee VALUES (101, '2019-01-01', 'EmpA', 10000, 'M', 'empA@email.com', 'Manager',
1);
INSERT INTO Employee VALUES (102, '2018-02-01', 'EmpB', 8000, 'F', 'empB@email.com', 'Assistant', 2);
INSERT INTO Employee VALUES (103, '2019-05-01', 'EmpC', 9000, 'M', 'empC@email.com', 'Clerk', 3);
```

PhoneNumbers Table

```
CREATE TABLE PhoneNumbers (
    PhoneNumberID NUMBER PRIMARY KEY,
    EntityID NUMBER,
    EntityType VARCHAR2(10) CHECK (EntityType IN ('LandOwner', 'Employee')),
    PhoneNumber VARCHAR2(15)
);
INSERT INTO PhoneNumbers VALUES (1, 1, 'LandOwner', '123-456-7890');
INSERT INTO PhoneNumbers VALUES (2, 2, 'LandOwner', '234-567-8901');
INSERT INTO PhoneNumbers VALUES (3, 3, 'LandOwner', '345-678-9012');
INSERT INTO PhoneNumbers VALUES (4, 101, 'Employee', '456-789-0123');
INSERT INTO PhoneNumbers VALUES (5, 102, 'Employee', '567-890-1234');
```

```
INSERT INTO PhoneNumbers VALUES (6, 103, 'Employee', '678-901-2345');
```

Grave Table

```
CREATE TABLE Grave (
   LandCode NUMBER PRIMARY KEY,
   Location VARCHAR2(100),
   size NUMBER(10,2)
);
INSERT INTO Grave VALUES (1, 'NorthSection', 15.5);
INSERT INTO Grave VALUES (2, 'EastSection', 12.0);
INSERT INTO Grave VALUES (3, 'WestSection', 14.5);
```

----------End----------