Jadavpur University

Database Management System Lab Report

Class: BCSE UG III

NAME-TONMOY BISWAS

ROLL – 002110501133

Section: A3

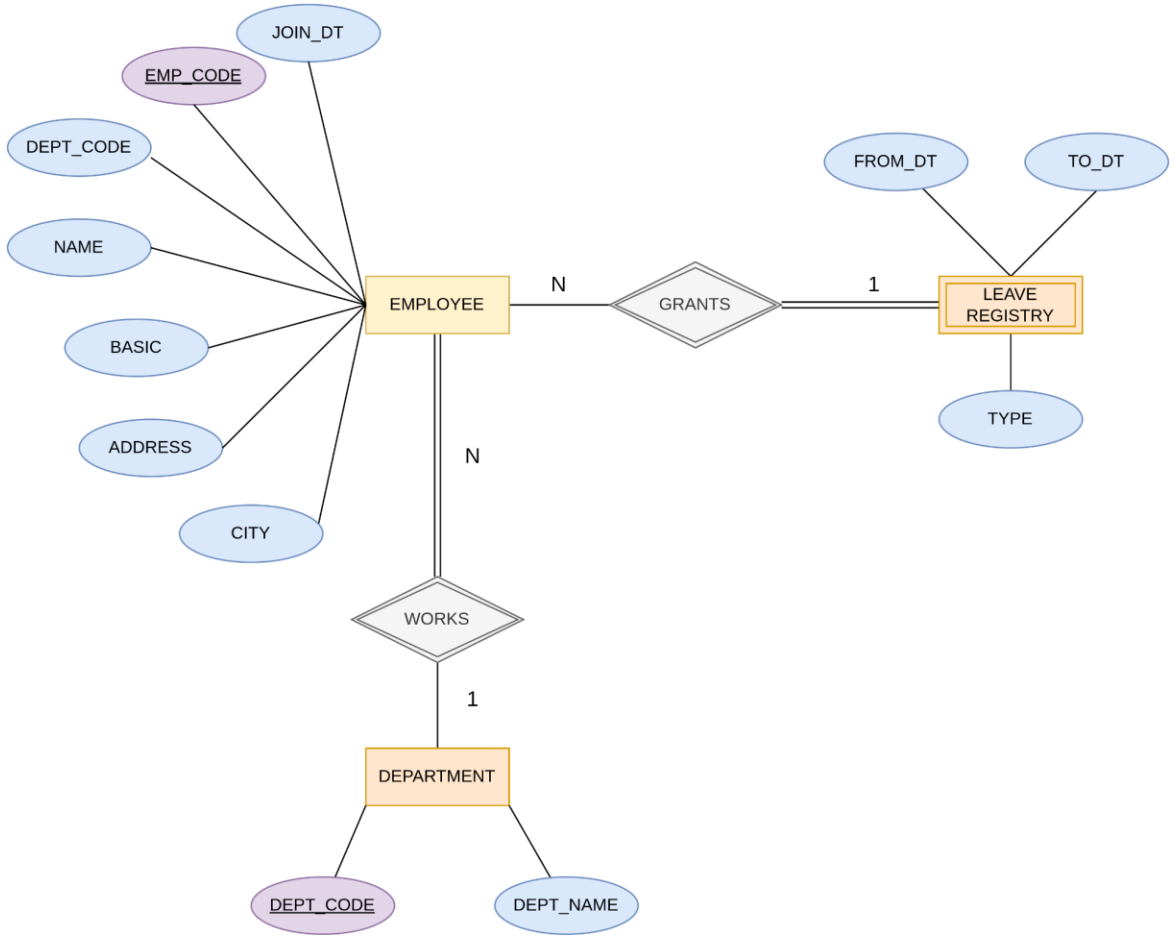
Assignment: 3

Semester: 6th

Session-: 2023-2024

1. In an organization, a number of departments exist. Each department has a name & unique code. Number of employees work in each department. Each employee has a unique employee code. Detailed information like name, address, city, basic, date of join are also stored. In a leave register for each employee leave records are kept showing leave type (CL/EL/ML etc.), from-date, to-date. When an employee retires or resigns then all the leave information pertaining to him are also deleted. Basic salary must be within Rs.5000 to Rs.9000. A department can not be deleted if any employee record refers to it. Valid grades are A/B/C. Employee name must be in uppercase only. Default value for joining date is system date. Design & implement the tables with necessary constraints to support the scenario depicted above.

ER Diagram



SQL For Create Table

```
CREATE TABLE DEPARTMENT(
```

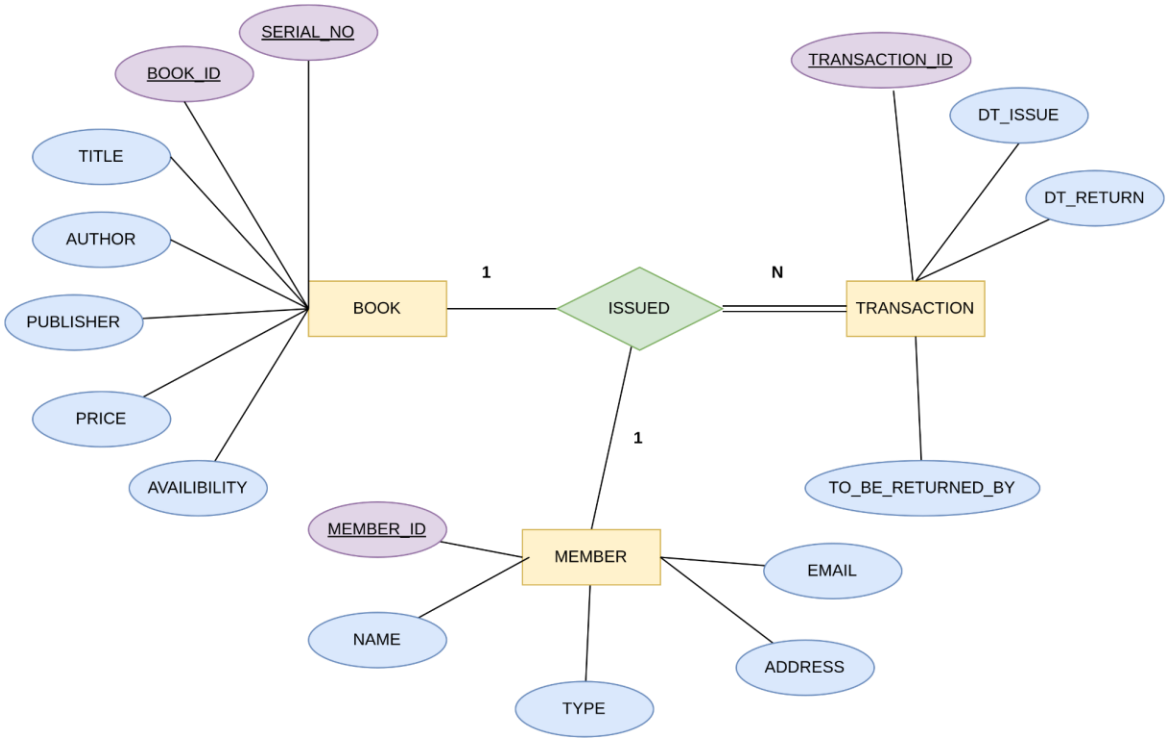
```
  DEPTCODE VARCHAR(10) PRIMARY KEY,
  NAME VARCHAR(15) NOT NULL
);

CREATE TABLE EMPLOYEE(
  NAME VARCHAR(25) NOT NULL ,
  EMPCODE CHAR(5) PRIMARY KEY,
  ADDRESS VARCHAR(50) NOT NULL,
  CITY VARCHAR(20) NOT NULL,
  BASIC INTEGER NOT NULL ,
  GRADE CHAR(1) NOT NULL ,
  JN_DATE DATE DEFAULT(CURRENT_DATE),
  DEPTCODE VARCHAR(10) NOT NULL,
  FOREIGN KEY(DEPTCODE) REFERENCES DEPARTMENT(DEPTCODE) ON DELETE RESTRICT
  CONSTRAINT NAME_UPPERCASE_CONSTRAINT CHECK(NAME = UPPER(NAME)),
  CONSTRAINT BASIC_RANGE_CONSTRAINT CHECK(BASIC>=5000 AND BASIC<=9000),
  CONSTRAINT GRADE_CONSTRAINT CHECK(GRADE IN ('A','B','C'))
);

CREATE TABLE LEAVE_REGISTER(
  EMPCODE CHAR(5) NOT NULL,
  LEAVE_TYPE CHAR(5) NOT NULL,
  FROM_DATE DATE NOT NULL,
  TO_DATE DATE NOT NULL,
  PRIMARY KEY(FROM_DATE,EMPCODE),
  FOREIGN KEY(EMPCODE) REFERENCES EMPLOYEE(EMPCODE) ON DELETE CASCADE,
  CONSTRAINT TYPE_CONSTRAINT CHECK(LEAVE_TYPE IN ('CL', 'EL', 'ML'))
);
```

4. In a library, for each book book-id, serial number (denotes copy number of a book), title, author, publisher and price are stored. Book-id and serial number together will be a unique identifier for a book. Members are either students or faculty. Each member has a unique member-id. Name, e-mail, address are also to be stored. Maximum number of books that a member can retain depends on the member type. There may be other such parameters that depend on member type. Design should be flexible. For any transaction (book issue or return), members are supposed to place transactions slip. Each Transaction will have a unique id. Users will submit member-id, book-id, and serial number (only for book return). Design and create the tables to store the book, member and transaction information. When a book is issued to a member of a field like, To_Be_Returned_By has to be set as DT_Issue + 7 days. At the time of book return, DT_Return will store the actual return date. While a new book arrives, the serial number will be the last serial number for the Book-id +1. System should also keep track of the status of each physical book -- whether issued or available.

ER Diagram



CREATE TABLE

```
-- Create tables
CREATE TABLE book(
    book_id INT NOT NULL,
    serial_num INT NOT NULL,
    title VARCHAR(100),
    author VARCHAR(50),
    publisher VARCHAR(60),
    price INT,
    available BOOLEAN DEFAULT true,
    PRIMARY KEY (book_id, serial_num)
);


CREATE TABLE member(
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(25),
    email VARCHAR(100),
    member_type CHAR(7),
    max_books INT,
    CONSTRAINT check_type CHECK(member_type IN ('faculty', 'student')),
    CONSTRAINT check_maxbooks CHECK((member_type = 'faculty' AND
max_books = 10) OR (member_type = 'student' AND max_books = 4))
);

CREATE TABLE transaction_slip(
    id INT PRIMARY KEY AUTO_INCREMENT,
    member_id INT,
```

```
    book_id INT,
    book_serial INT,
    issue_date DATE DEFAULT(CURRENT_DATE),
    return_date DATE,
    to_be_returned_by DATE,
    FOREIGN KEY (member_id) REFERENCES member(id) ON DELETE RESTRICT,
    FOREIGN KEY (book_id, book_serial) REFERENCES book(book_id,
serial_num) ON DELETE RESTRICT
);
```