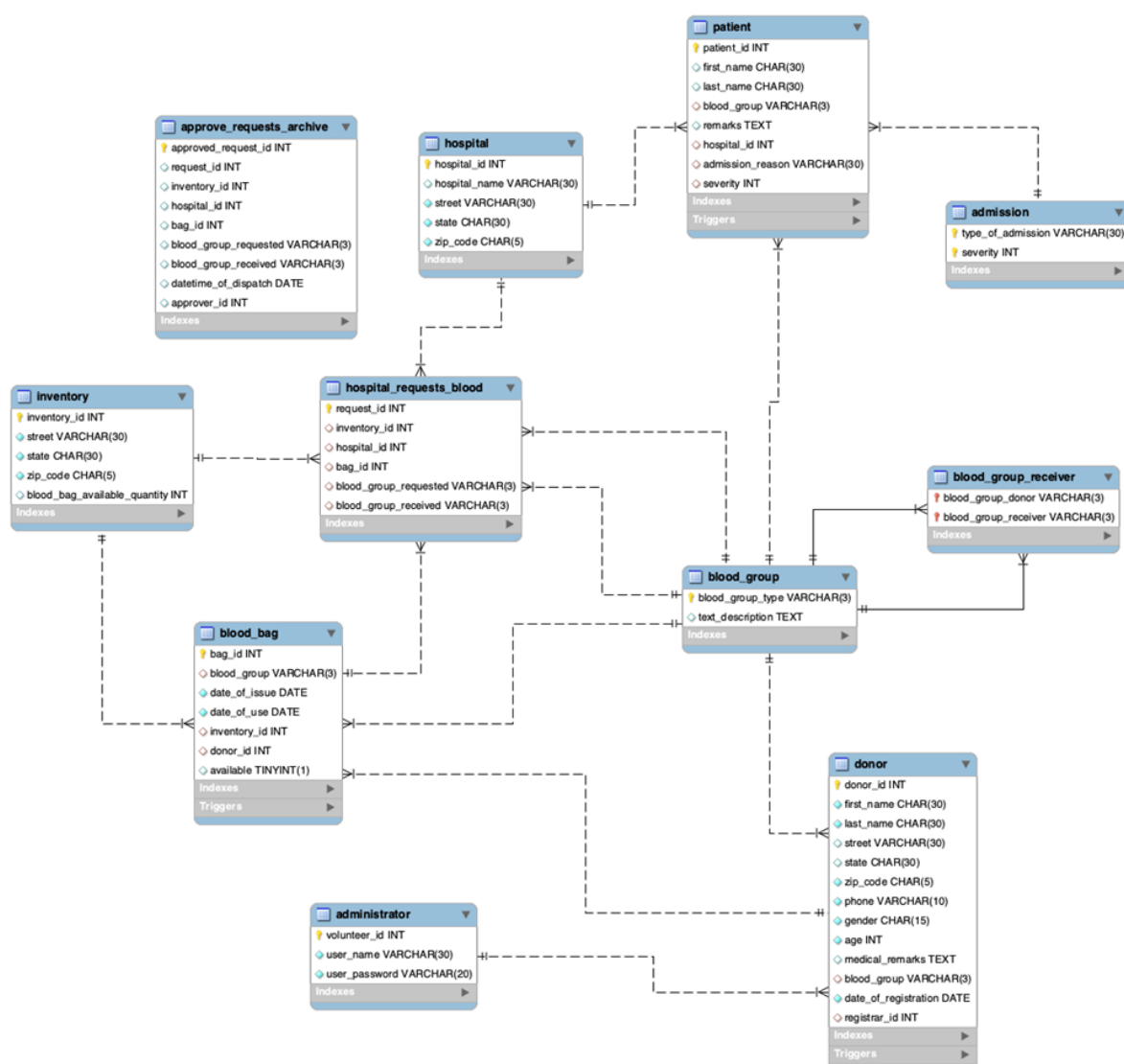


CS 5200 Hands-on Exercise

[50pt] Database Implementation



1) [25pt] SQL queries

- For each table,

a) **[15pt]** Must present **screenshots of your SQL query** that creates a table for each table

- First, you need to define and present your own decision for enforcing the minimum cardinality for each relationship
- You must define and implement at least three data constraints.
- Your queries must consider all required data constraints and table constraints that are defined in the design.

b) **[10pt]** Must present **screenshots of your SQL query** that inserts data entries for each table

(Weak/Strong) Max: Min - Min: Max

2) **[25pt]** Result of Implementation

Please submit an **exported database**.

* The minimum number of rows is **10**.

Table creating part:

TODO:

- modify the self-increment initial values.
- Add some constraints if possible

```
CREATE DATABASE bds;  
CREATE TABLE administrator (
```

```

VolunteerID Int NOT NULL AUTO_INCREMENT,
UserName VARCHAR(30) NOT NULL,
UserPassword VARCHAR(20) NOT NULL,
CONSTRAINT PK_ADMN PRIMARY
KEY(VolunteerID),
CONSTRAINT AK_ADMN UNIQUE(UserName)
)AUTO_INCREMENT = 1;

CREATE TABLE blood_group (
    BloodGroupType VARCHAR(3) NOT NULL,
    TextDescription TEXT DEFAULT NULL,
    CONSTRAINT PK_BG PRIMARY
KEY(BloodGroupType),
    CONSTRAINT BG_Domain CHECK (BloodGroupType
IN ('A+', 'A-', 'B+', 'B-', 'O+', 'O-', 'AB+',
'AB-')) )
);

CREATE TABLE blood_group_receiver (
    BloodGroupDonor VARCHAR(3) NOT NULL,
    BloodGroupReceiver VARCHAR(3) NOT NULL,
    CONSTRAINT PK_BGR PRIMARY KEY
(BloodGroupDonor, BloodGroupReceiver),
    CONSTRAINT BG_FK1 FOREIGN KEY
(BloodGroupDonor) REFERENCES
blood_group(BloodGroupType)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT BG_FK2 FOREIGN KEY
(BloodGroupReceiver) REFERENCES
blood_group(BloodGroupType)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION

```

);

```
CREATE TABLE donor ( -- add trigger
    DonorID INT NOT NULL AUTO_INCREMENT,
    FirstName CHAR(30) NOT NULL,
    LastName CHAR(30) NOT NULL,
    Street VARCHAR(30) DEFAULT NULL,
    State CHAR(30) DEFAULT NULL,
    ZipCode CHAR(5) NOT NULL,
    Phone VARCHAR(10) NOT NULL,
    Gender CHAR(15) NOT NULL,
    Age Int NOT NULL,
    MedicalRemarks TEXT DEFAULT NULL,
    BloodGroup VARCHAR(3) NOT NULL,
    DateOfRegistration DATE NOT NULL,
    RegistrarID Int NOT NULL,
    CONSTRAINT PK_DNR PRIMARY KEY (DonorID),
    CONSTRAINT AK_DNR UNIQUE(Phone),
    CONSTRAINT valid_Age CHECK (Age > 0 AND Age
< 100),
    CONSTRAINT valid_ZipCode CHECK (
    ZipCode LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT valid_Phone CHECK (
    Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9]
[0-9][0-9][0-9][0-9]'),
    CONSTRAINT BG_FK FOREIGN KEY (BloodGroup)
REFERENCES blood_group(BloodGroupType)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT ADMN_FK FOREIGN KEY
(RegistrarID) REFERENCES
administrator(VolunteerID)
    ON UPDATE NO ACTION
```

```

        ON DELETE NO ACTION
    ) AUTO_INCREMENT = 1;

CREATE TABLE inventory (
    InventoryID INT NOT NULL AUTO_INCREMENT,
    Street VARCHAR(30) NOT NULL,
    State CHAR(30) NOT NULL,
    ZipCode CHAR(5) NOT NULL,
    BloodBagQtn Int NULL,
    CONSTRAINT PK_INVNTRY PRIMARY
KEY(InventoryID),
    CONSTRAINT valid_ZipCode CHECK (ZipCode
LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CONSTRAINT valid_BloodBagQtn CHECK (
        BloodBagQtn > -1)
) AUTO_INCREMENT = 1;

CREATE TABLE blood_bag ( -- add trigger
    BagID INT NOT NULL AUTO_INCREMENT,
    BloodGroup VARCHAR(3) NOT NULL,
    DateOfIssue DATE NOT NULL,
    DateOfuse DATE NOT NULL,
    InventoryID INT NOT NULL,
    DonorID INT NOT NULL,
    Available TINYINT(1) NOT NULL DEFAULT 1,
    CONSTRAINT PK_BB PRIMARY KEY(BagID),
    CONSTRAINT FK1_BB FOREIGN KEY (BloodGroup)
REFERENCES blood_group(BloodGroupType)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT FK2_BB FOREIGN KEY (InventoryID)
REFERENCES inventory(InventoryID)
    ON UPDATE NO ACTION

```

```
    ON DELETE NO ACTION,  
    CONSTRAINT FK3_BB FOREIGN KEY (DonorID)  
REFERENCES donor(DonorID)  
    ON UPDATE NO ACTION  
    ON DELETE NO ACTION,  
    CONSTRAINT Date_Check CHECK (DateOfuse >  
DateOfIssue),  
    CONSTRAINT valid_Available CHECK (Available  
= 1 OR Available = 0)  
) AUTO_INCREMENT = 1;
```

```
CREATE TABLE hospital (  
    HospitalID INT NOT NULL AUTO_INCREMENT,  
    HospitalName VARCHAR(30) NULL,  
    Street VARCHAR(30) NOT NULL,  
    State CHAR(30) NOT NULL,  
    ZipCode CHAR(5) NOT NULL,  
    CONSTRAINT PK_HSPTL PRIMARY  
KEY(HospitalID),  
    CONSTRAINT AK_HSPTL UNIQUE(HospitalName),  
    CONSTRAINT valid_ZipCode CHECK (ZipCode  
LIKE '[0-9][0-9][0-9][0-9][0-9]'),  
) AUTO_INCREMENT = 1;
```

```
CREATE TABLE hospital_requests_blood (  
    RequestID INT NOT NULL AUTO_INCREMENT,  
    InventoryID INT NOT NULL,  
    HospitalID INT NOT NULL,  
    BagID INT NOT NULL,  
    BloodGroupReq VARCHAR(3) NOT NULL,  
    BloodGroupRec VARCHAR(3) NOT NULL,  
    CONSTRAINT PK_HRB PRIMARY KEY(RequestID),
```

```

        CONSTRAINT BG_FK1 FOREIGN KEY
(BloodGroupReq) REFERENCES
blood_group(BloodGroupType)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
        CONSTRAINT BG_FK2 FOREIGN KEY
(BloodGroupRec) REFERENCES
blood_group(BloodGroupType)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
        CONSTRAINT INVNTY_FK1 FOREIGN KEY
(InventoryID) REFERENCES inventory(InventoryID)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
        CONSTRAINT HSPTL_FK1 FOREIGN KEY
(HospitalID) REFERENCES hospital(HospitalID)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
        CONSTRAINT BB_FK1 FOREIGN KEY (BagID)
REFERENCES blood_bag(BagID)
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
) AUTO_INCREMENT = 1;

```

```

CREATE TABLE admission (
    TypeOfAdmission VARCHAR(30) NOT NULL,
    Serverity INT NOT NULL AUTO_INCREMENT,
    CONSTRAINT PK_ADM PRIMARY KEY
(TypeOfAdmission, Serverity),
    CONSTRAINT valid_Serverity CHECK (Serverity
>= 0 AND Serverity < 7) -- AIS
) AUTO_INCREMENT = 1;

```

```

CREATE TABLE patient ( -- add trigger
    PatientID INT NOT NULL AUTO_INCREMENT,
    FirstName CHAR(30) NULL,
    LastName CHAR(30) NULL,
    BloodGroup VARCHAR(3) NOT NULL,
    Remarks TEXT DEFAULT NULL,
    HospitalID Int NOT NULL,
    AdmissionReason VARCHAR(30) NOT NULL,
    Serverity Int NOT NULL,
    CONSTRAINT PK_PTNT PRIMARY KEY(PatientID),
    CONSTRAINT BG_FK1 FOREIGN KEY (BloodGroup)
REFERENCES blood_group(BloodGroupType)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT ADM_FK1 FOREIGN KEY
(AdmissionReason) REFERENCES
admission(TypeOfAdmission)
    ON CASCADE UPDATE
    ON CASCADE DELETE,
    CONSTRAINT HSPTL_FK1 FOREIGN KEY
(HospitalID) REFERENCES hospital(HospitalID)
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT ADM_FK2 FOREIGN KEY (Serverity)
REFERENCES admission(Serverity)
    ON CASCADE UPDATE
    ON CASCADE DELETE
) AUTO_INCREMENT = 1;

```

```

CREATE TABLE approve_requests_archive (
    ApprovedRequestID INT NOT NULL
    AUTO_INCREMENT,
    RequestID INT NULL,

```



```

InventoryID INT NULL,
HospitalID INT NULL,
BagID INT NULL,
BloodGroupReq VARCHAR(3) NULL,
BloodGroupRec VARCHAR(3) NULL,
DatetimeOfDispatch Date NULL,
ApproverID INT NULL,
CONSTRAINT PK_ARA PRIMARY
KEY(ApprovedRequestID)
) AUTO_INCREMENT = 1;

```

Trigger Creating part:

TODO:

- finish the trigger

```

DELIMITER //
CREATE TRIGGER donor_initail_value
BEFORE INSERT ON donor
FOR EACH ROW
-- I am thinking if a donor come with the
bloodtype not in the bloodgroup shall I also
create a new entry in bloodgroup, or I shall
think all possible bloodgroups already stores
in the bloodgroup table?
BEGIN
    DECLARE VarVolunteerID INT;
    DECLARE Flag1 INT DEFAULT 0;
    DECLARE Flag2 INT DEFAULT 0;
    DECLARE VarBloodGroupType VARCHAR(3);
    SET VarVolunteerID = NEW.VolunteerID;
    SET VarBloodGroupType = NEW.BloodGroupType;

```

```

SELECT COUNT(VolunteerID) INTO FLag1
FROM administrator
WHERE VolunteerID = VarVolunteerID

SELECT COUNT(VarBloodGroupType) INTO FLag2
FROM blood_group
WHERE BloodGroupType = VarVolunteerID

IF (FLag1 = 0 OR FLAG2 = 0) THEN
    SIGNAL SQLSTATE '45000' SET
MESSAGE_TEXT = 'Admin or blood group invalid';

END IF;

END

//
DELIMITER ;

DELIMITER ;

DELIMITER //
CREATE TRIGGER blood_bag_initail_value
BEFORE INSERT ON blood_bag
FOR EACH ROW

BEGIN
    DECLARE VarDonorID INT;
    DECLARE VarBloodGroupType VARCHAR(3),
    DECLARE VarInventoryID INT;
    DECLARE FLag INT;

```

```

SET VarDonorID = NEW.DonorID;
SET VarInventoryID = NEW.InventoryID;
SET VarBloodGroupType = NEW.BloodGroupType;

SELECT SUM(COUNT(DonorID),
COUNT(BloodGroupType), COUNT(InventoryID)) INTO
Flag
FROM donor, blood_group, inventory
WHERE DonorID = VarDonorID AND
VarBloodGroupType = BloodGroupType AND
VarInventoryID = InventoryID;

IF (Flag < 3) THEN
    SIGNAL SQLSTATE '45000' SET
MESSAGE_TEXT = 'donor or blood group or
inventory invalid';
END IF;

END

//
DELIMITER ;

DELIMITER //
CREATE TRIGGER patient_initail_value
BEFORE INSERT ON patient
FOR EACH ROW

BEGIN
    DECLARE VarHospitalID INT;

```

```

DECLARE VarBloodGroupType VARCHAR(3),
DECLARE VarSeverity INT;
DECLARE VarReason VARCHAR(30);
DECLARE Flag INT;

SET VarHospitalID = NEW.HospitalID;
SET VarReason = NEW.TypeOfAdmission;
SET VarSeverity = NEW.Severity;
SET VarBloodGroupType = NEW.BloodGroupType;

SELECT SUM(COUNT(HospitalID),
COUNT(BloodGroupType), COUNT(TypeOfAdmission),
COUNT(Severity)) INTO Flag
FROM hospital, blood_group, admission
WHERE HospitalID = VarHospitalID AND
VarBloodGroupType = BloodGroupType AND
TypeOfAdmission = VarReason, Severity =
VarSeverity;

IF (Flag < 4) THEN
    SIGNAL SQLSTATE '45000' SET
MESSAGE_TEXT = 'donor or blood group or
inventory invalid';
END IF;

END

//
DELIMITER ;

```

