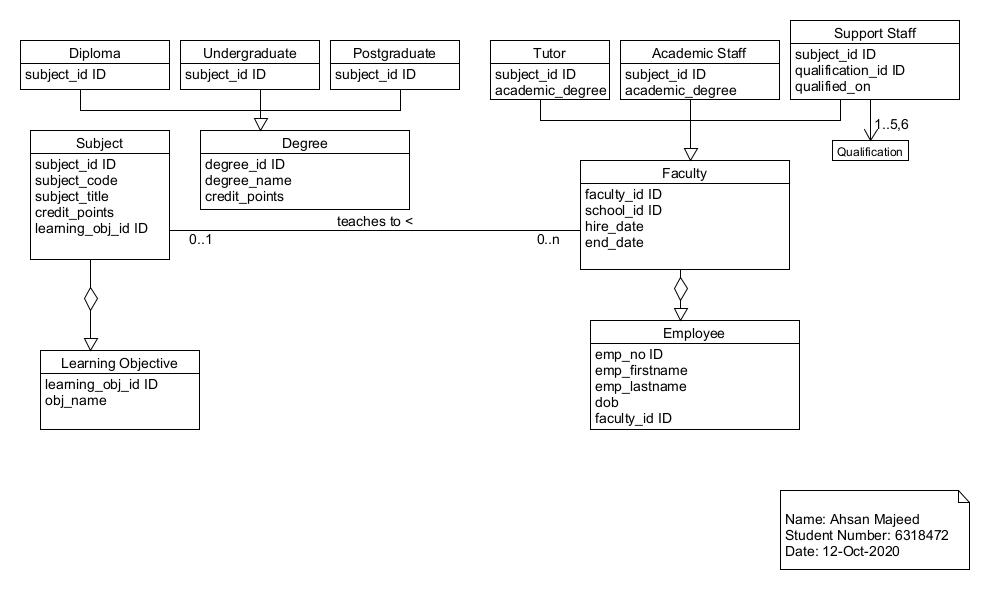
**Question 1**

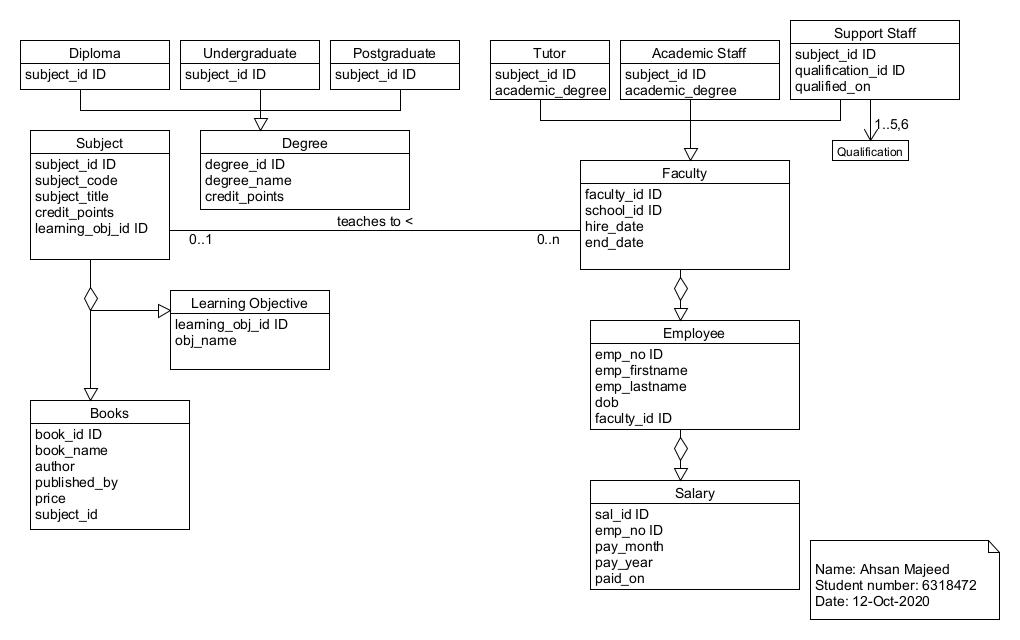
1.Draw a conceptual schema for the specification of the database domain listed above. Use the UML simplified class diagrams explained during this subject. Note: you are not allowed to use any artificial identifiers (create new IDs) and or add any attributes that are not explicitly mentioned in the specification.

Answer 1.

****

***Figure 1. ER Diagram of University Management***

2.Add two (2) new object classes with at least five (5) attributes each and appropriate associations. The choice of object classes, attributes and associations are up to you; however, these should relate to the existing scenario.

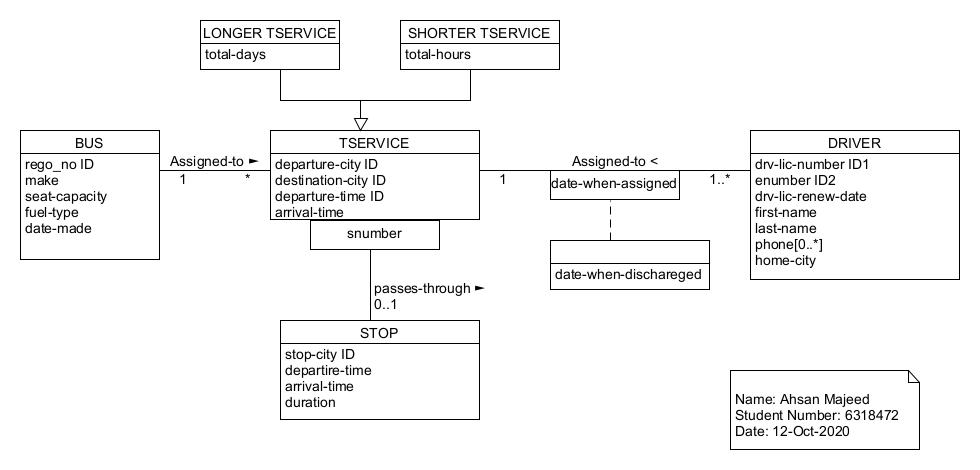


***Figure 2. Updated ER diagram of University Management***

**Question 2**

Your task is to perform the steps of logical database design, i.e. to transform a conceptual schema given above into a collection of relational schemas. Before transforming add the attribute ‘date-made’ to the BUS class and ‘home-city’ to the DRIVER class. Draw the resulting conceptual schema adding your name, student number and the date to the drawing. Use UMLet and paste images of your drawings into your Microsoft Word document using the template provided. List the names of attributes, primary key, candidate keys (if any), and foreign keys (if any) for each class in the relational schema. Assume that an association method is used to implement the generalization. Show your working and explain as you step through the transformation process.

Answer 2.

****

***Figure 1. Modified ER Diagram of Conceptual Schema***

Added **date-made** field in BUS class and **home-city** field in DRIVER class.

In order to transform a conceptual schema given above into a collection of relational schemas, following steps are performed in a given order below:

Fisrstly I have created a table named as 'TSERVICE' having fields - service number which is the primary key of table

along with the departure city and destination city id and departure and arrival time.

CREATE TABLE TSERVICE(

TSERVICENUM DECIMAL(12) NOT NULL,

DEPARTURE\_CITY DECIMAL(12) NOT NULL,

DESTINATION\_CITY DECIMAL(12))NOT NULL,

DEPARTURE\_TIME VARCHAR(50) NOT NULL,

ARRIVAL\_TIME VARCHAR(50) NOT NULL,

CONSTRAINT TSERVICE\_PKEY PRIMARY KEY(TSERVICENUM) );

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

Then service table is further normalized ino longer trip service and shorter trip services and which stores the total days in longer services while total hours in shorter services

respectively. Both the tables has primary key of TSERVICE table as foreign key for reference.

CREATE TABLE LONGER\_TSERVICE(

TLONGER\_SERVICENUM DECIMAL(12) NOT NULL,

TSERVICENUM DECIMAL(12) NOT NULL,

TOTAL\_DAYS VARCHAR(50) NOT NULL,

CONSTRAINT LONGER\_TSERVICE\_PKEY PRIMARY KEY(TLONGER\_SERVICENUM),

CONSTRAINT LONGER\_TSERVICE\_FKEY FOREIGN KEY(TSERVICENUM) REFERENCES TSERVICE(TSERVICENUM) );

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

CREATE TABLE SHORTER\_TSERVICE(

TSHORTER\_SERVICENUM DECIMAL(12) NOT NULL,

TSERVICENUM DECIMAL(12) NOT NULL,

TOTAL\_HOURS VARCHAR(50) NOT NULL,

CONSTRAINT SHORTER\_TSERVICE\_PKEY PRIMARY KEY(TSHORTER\_SERVICENUM),

CONSTRAINT SHORTER\_TSERVICE\_FKEY FOREIGN KEY(TSERVICENUM) REFERENCES TSERVICE(TSERVICENUM) );

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

Further, each trip is carried out by a driver so we are maintaining individual DRIVER table in order to record the basic information about the driver.

CREATE TABLE DRIVER(

DRI\_LIC\_NUMBER DECIMAL(12) NOT NULL,

ENUMBER DECIMAL(8) NOT NULL,

DRI\_LIC\_RENEW\_DATE DATE NOT NULL,

FIRSTNAME VARCHAR(50),

LASTNAME VARCHAR(50),

PHONE DECIMAL(12) NOT NULL,

HOME\_CITY VARCHAR(50),

CONSTRAINT DRIVER\_PKEY PRIMARY KEY(ENUMBER),

CONSTRAINT DRIVER\_UNIQUE UNIQUE(DRI\_LIC\_NUMBER));

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

Along with the driver, buses information are also stored in separate table.

CREATE TABLE BUS(

REGO\_NO DECIMAL(12) NOT NULL,

MAKE VARCHAR(50),

SEAT\_CAPACITY DECIMAL(12) NOT NULL,

FUEL\_TYPE VARCHAR(50) NOT NULL,

DATE\_MADE DATE NOT NULL,

CONSTRAINT BUS\_PKEY PRIMARY KEY(REGO\_NO));

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

I have also created a separate STOP table as mentioned in the conceptual schema for storing stoppage related information of the trips.

CREATE TABLE STOP(

STOP\_ID DECIMAL(12) NOT NULL,

STOP\_CITY DECIMAL(12) NOT NULL,

DEPARTURE\_TIME VARCHAR(50) NOT NULL,

ARRIVAL\_TIME VARCHAR(50) NOT NULL,

DURATION VARCHAR(50) NOT NULL,

CONSTRAINT STOP\_PKEY PRIMARY KEY(STOP\_ID));

/\* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ \*/

**Question 3**

**1.)**

**- First Delete the constraint and add with new configuration.**

**ALTER TABLE TRUCK DROP CONSTRAINT TRUCK\_CAPACITY;**

**ALTER TABLE TRUCK ADD CONSTRAINT TRUCK\_CAPACITY CHECK (CAPACITY>0.0 AND CAPACITY<=200)**

**2.)**

**- First add the new column and then add the constraint to that column.**

**ALTER TABLE TRIPLEG ADD COLUMN TOTAL\_LEGS INT(2);**

**ALTER TABLE TRIPLEG ADD CONSTRAINT LIMIT\_TOTAL\_LEGS CHECK(TOTAL\_LEGS>0 AND TOTAL\_LEGS<=10);**

**3.)**

**- Please note that we can also ignore fields like FIRSTNAME, FIRSTNAME, DOB because every mechanic is an employee and we can get this info from employee table by joining with employeenum.**

**(But, for understanding, I have added the fields)**

**- Here, employeenum will be foreign key as mechanic is an employee and without employee, mechanic cannot exist.**

**CREATE TABLE aman.MECHANICS (**

**EMPLOYEENUM DECIMAL(12) NOT NULL,**

**FIRSTNAME VARCHAR(50) NOT NULL,**

**LASTNAME VARCHAR(50) NOT NULL,**

**DOB DATE NULL,**

**QUALIFICATIONLEVEL INT(1) NOT NULL,**

**CONSTRAINT MECHANIC\_ID FOREIGN KEY(EMPLOYEENUM) REFERENCES EMPLOYEE (EMPLOYEENUM),**

**CONSTRAINT QUAL\_LEVEL CHECK (**

**QUALIFICATIONLEVEL > 0**

**AND QUALIFICATIONLEVEL <= 5**

**)**

**);**

4.)

- In this case, we need to remove the foreign key constraint from trip table. It is so because, if license number is foreign key, we have to delete the trip info when a driver info is deleted. So, to resolve this, the foreign key constraint must be dropped.

**ALTER TABLE Trip**

**DROP CONSTRAINT TRIP\_FK1; -- Drops the foreign key on license number**

**Question 4**

**First insert the values for the given tables using query insertinto**

**1) Select \* from trip where departure='sydney' and destination='wollongong' and licence num='432908';**

**2) Delete \* from trip**

**where tripnum='60';**

**3) Update driver**

**Set status ='onLeave'**

**where tripnum>110;**

**4) Select \* into E20cent**

**from employees**

**where dob<2000;**

**Question 5**

**(1.)**

**select e.firstname,e.lastname from employee as e,driver as d where d.employeenum=e.employeenum and d.status='ON LEAVE'**

**(2.)**

**select t.regnum as "Registration No." from truck as t,trip as p where t.regnum=p.regnum**

**(3.)**

**select t.regnum as "Registration No.",count(p.tripnum) as "No. of times used " from truck as t,trip as p where t.regnum=p.regnum group by t.regnum having count(p.tripnum) < 80**

**(4.)**

**(select departure from tripleg where tripnum=7 or tripnum=11) union (select destination from tripleg where tripnum=7 or tripnum=11)**

**(5.)**

**select t.regnum as "Registration No." from truck as t,trip as p where t.regnum=p.regnum and p.licensenum<>306234 and p.licensenum<>901108**