

Name	Nachiket Rao
Student ID	S3827657
Tutorial: Day, Time , Location	Tuesday, 10.30-12.30, 014.09.015
Tutor Name	Kenneth Gardiner

DATABASE CONCEPTS

Assignment 1

Question 1 - The Relational Model

1.1 – Schema

STUDENT (sno, surname, givenname, major)

STAFF (eno, surname, givenname, department)

CLASS (cno, lecturer*, day, time, room)

ENROL (sno*, eno*, grade)

- Lecturer column is a foreign key referencing the primary key eno from STAFF table

1.2 – Create Table statements

```
CREATE TABLE student
(
    sno          VARCHAR(1000) NOT NULL,
    surname      VARCHAR(100),
    givenname    VARCHAR(100) NOT NULL,
    major        VARCHAR(100),
    PRIMARY KEY (sno)
);

CREATE TABLE staff
(
    eno          VARCHAR(1000) NOT NULL,
    surname      VARCHAR(100),
    givenname    VARCHAR(100) NOT NULL,
    department   VARCHAR(1000),
    rank         VARCHAR(100),
    PRIMARY KEY (eno)
);

CREATE TABLE class
(
    cno          VARCHAR(1000) NOT NULL,
    lecturer     VARCHAR(1000),
    day          VARCHAR(100),
    TIME         VARCHAR(5),
    room         VARCHAR(100),
    PRIMARY KEY (cno),
    CONSTRAINT fk_class_staff FOREIGN KEY (lecturer) REFERENCES staff(eno)
);

CREATE TABLE enrol
(
    sno          VARCHAR(1000) NOT NULL,
    cno          VARCHAR(1000) NOT NULL,
    grade        VARCHAR(100),
    PRIMARY KEY (cno, sno),
    CONSTRAINT fk_enrol_student FOREIGN KEY (sno) REFERENCES student(sno),
```

```
        CONSTRAINT fk_enrol_class FOREIGN KEY (cno) REFERENCES class(cno)
    );
```

1.3 – Insert Queries

```
insert into Student values('s1001','Smith','Tom','History');
```

```
insert into Student values('s1002','Chin','Ann','Maths');
```

```
insert into Student values('s1003','Lee','Perry','Arts');
```

```
insert into Student values('s1005','Smith','John','History');
```

```
insert into Student values('s1006','River','Jane','Arts');
```

```
insert into Staff values('e123','Bowl','Alex','Maths','Lecturer');
```

```
insert into Staff values('e205','Cox','Kevin','CSC','Associate Professor');
```

```
insert into Staff values('e301','Jones','David','Arts','Senior Lecturer');
```

```
insert into Class values('isys155','e123','Wed','17:30','80.01.12');
```

```
insert into Class values('cosc121','e205','Thu','08:30','12.10.02');
```

```
insert into Class values('artc131','e301','Mon','10:30','10.08.09');
```

```
insert into Class values('cosc101','e205','Tue','14:30','14.09.05');
```

```
insert into Enrol values('s1001','isys155','HD');
```

```
insert into Enrol values('s1003','cosc121',null);
```

```
insert into Enrol values('s1005','artc131','CR');
```

```
insert into Enrol values('s1006','cosc101',null);
```

Question 2 – SQL

2.1 – Query explanation

This query fetches the field number along with the field name from the field table of those records whose title either starts with the word 'data', ends with 'data' or the title contains data somewhere in its string. The query will also fetch records whose field number lies in the range of 500 to 599 inclusive.

2.2 – Correct SQL Query

```
SELECT DISTINCT panum,
               title,
               acnum,
               fieldnum
FROM   paper
      INNER JOIN author using (panum)
      INNER JOIN interest using (acnum);
```

2.3 - How many academics are there in the department where deptnum=100? Return the total number.

```
SELECT Count(acnum)
FROM   academic
WHERE  deptnum = 100;

/*this query fetches the number of
academics whose department number is
100 and the result is 17*/
```

2.4 - List the titles of all papers in the database, in alphabetical order.

```
SELECT title
FROM   paper
ORDER  BY title;

/*this query fetches the titles
Of all papers in the database and
Orders them alphabetically from a to z*/
```

2.5 - Return the details of research fields which have a title starting with the word "Data". Note that the result should include the fields "Data" or "Data Structures" but not "Databases".

```
SELECT *
FROM   field
WHERE  Upper(title) LIKE 'DATA %'
      AND Upper(title) NOT LIKE 'DATABASE%'

/*this query fetch all the columns
from field table where the title starts
with the word data AND does not start with
database. Case sensitivity eliminated by
```

converting the characters in title to upper
case before string comparison*/

2.6 - List the panum, title and author acnum of each paper.

```
SELECT panum,
       title,
       acnum
FROM   paper
       INNER JOIN author USING (panum)

/*this query performs an inner join on two table
i.e. paper and author using the foreign key panum
in author table which references the primary key
panum in paper table.*/
```

2.7 - Return the famname and givenname of academics working for 'RMIT CS' (descrip) with acnum in the range [200..299]. The output should be in alphabetical order of famname and then givenname.

```
SELECT famname,
       givenname
FROM   academic
       inner join department USING (deptnum)
WHERE  descrip = 'RMIT CS'
       AND ( acnum >= 200
            AND acnum <= 299 )
ORDER BY famname ASC,
       givenname ASC;

/*this query does an inner join on academic and
Department table using the deptnum attribute and returns
First and last name of those records with acnum between
200 to 299 inclusive and department name is RMIT CS*/
```

2.8 - List the famname, givenname of academics who work for institutions in Victoria. Note that the values for "Victoria" include "VIC" or "Vic"

```
SELECT famname,
       givenname
FROM   academic
       inner join department USING (deptnum)
WHERE  Upper(department.state) LIKE 'VIC%'
ORDER BY famname ASC,
       givenname ASC;

/*this query fetches the first and lastname
Of academics who work for institutes in the state
Of Victoria. % identifier is used as a placeholder
Include values containing VIC or Vic */
```

2.9 - Are there academics who do not have any title? Print their givenname, famname. The list should be in alphabetical order of famname and then givenname.

```
SELECT famname,  
       givenname  
FROM   academic  
WHERE  title IS NULL  
ORDER BY famname ASC,  
       givenname ASC
```

```
/*this query returns the first and last name  
Of records in the academic table where title  
Column has the value of null.  
*/
```

2.10 - How many institutions are there in the database?

```
SELECT count(DISTINCT instname)  
FROM   department
```

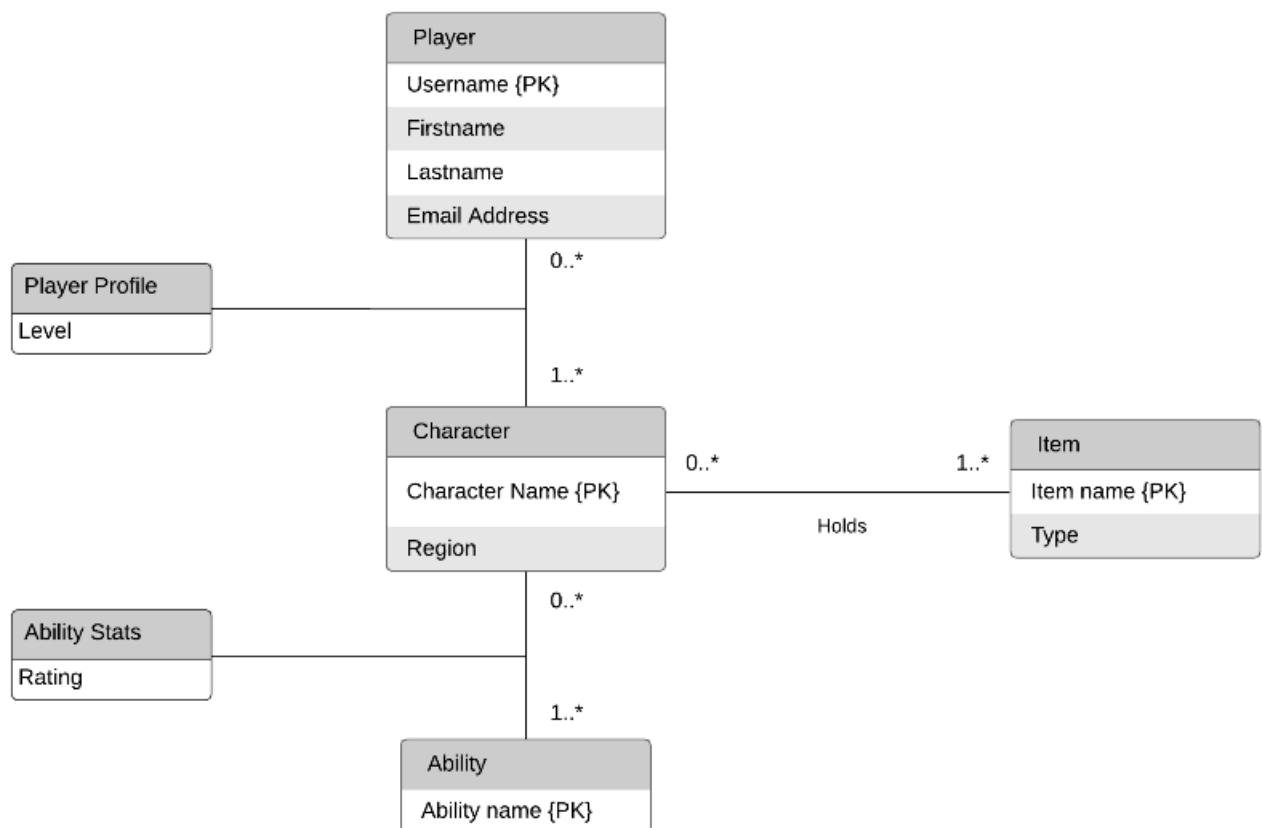
```
/*this query returns count of unique institution  
Names from the department.  
*/
```

Question 3 – ER Diagram

This diagram was made using Lucidchart software ^[1]

Assumptions:

- Region is not uniquely identified and cannot exist without a character. Additionally, since each character comes from a single region, region is taken as an attribute in the character entity
- Ability rating taken as a relation since each character can have one or more abilities and each ability will have a different rating for that character
- Level is a relation as a player can have different characters on different levels
- Item and abilities are entities since they are uniquely identified by item name and ability name respectively. Since they are entities, they can also exist independently hence the 0..* relationship



Unexplained Constraints/Ambiguities in Question description:

- The question assumes name to be a unique identifier for all entities. In the real-world scenario names are not used as primary keys due to the high probability of repetition
- It is unspecified whether a region can be developed without having a character associated with it. If yes, then region is an entity
- The description says that each character has a rating for each ability. However, it isn't stated whether this ability rating is upgradable depending on the player's progress as is with most games. In this case, rating will also be a relation associated with the player profile.

References:

[1] Lucidchart.com. 2020. [online] Available at: <<https://www.lucidchart.com/>> [Accessed 2 April 2020].