Question 1

A school's database looks like this (it was set up by someone more used to spreadsheets):

stuId	name	gender	unit	grade
101	Fred	M	Mathematics	75
101	Fred	M	German	65
101	Fred	M	English	90
102	Sam	X	Mathematics	60
102	Sam	X	English	60

stuId is a student id that is unique per student. Students' names are not required to be unique, i.e. you can have two 'Fred's in the school. Gender is one of M, F, X. For each student and each unit they take, there is one row containing among other things the student name, unit name and the grade (0-100) that the student got on this unit. In the example above, we can see that Fred took three units (Mathematics, German and English). No two units have the same name but a unit name can appear several times in the database since many students can take the same unit. The first row of the example tells us that there is a student called Fred with id 101, who is male, and took the Mathematics unit and got a grade of 75 on it.

(a) Identify the candidate key(s) in every table.

$$Relation = \{stdId, name, gender, unit, grade\}$$

$$Function = \left\{ \begin{array}{c} stuId \rightarrow name, \\ stuId \rightarrow gender, \\ stuId \rightarrow unit, \\ stuId\&unit \rightarrow grade \end{array} \right\}$$

In this way, we can get closure stuId:

$$\begin{aligned} (stuId)^+ &= \{stuId\} \\ &= \{stuId, name\} \quad (stdId \rightarrow name) \\ &= \{stuId, name, gender\} \quad (stuId \rightarrow gender) \\ &= \{stuId, name, gender, unit\} \quad (stuId \rightarrow unit) \\ &= \{stuId, name, gender, unit, grade\} \quad (stuId\&unit \rightarrow grade) \end{aligned}$$

$$So\ we\ get : (stuId)^+ \sqsupseteq Relation$$

So stuId is a candidate key.

(b) Identify the key and non-key attributes in every table.

Because stuLd is the only candidate key we get in this table, so stuId is candidate key and primary key.
Name, gender, unit, grade are non-key.

Question 1 continued on next page...

(c) Determine which normal forms from (1NF, 2NF, 3NF, BCNF) the schema does or does not satisfy. Give evidence to support your answer.

For function in this table:

$$Let: \ Function = F\_stuId \cap F\_grade$$

$$F\_stuId = \left\{ \begin{matrix} stuId \rightarrow name, \\ stuId \rightarrow gender, \\ stuId \rightarrow unit, \end{matrix} \right\}$$

stuId is candidate key, so F_stuId is satisfy in BCNF. In F_grade from Function, both unit and grade are non-key in Function. And unit relies on stdId in F_stuId.

$$F\_grade = \{stuId\&unit \rightarrow grade\}$$

In this way, both F_stuId and F_grade are satisfy in BCNF, but Function is not satisfy in BCNF.
Function is just satisfy in 1NF. For unit is non-key relying on stuId, however stuId&unit be a candidate key in a subset of Function.

(d) If the schema is not in BCNF, normalise it as far as possible(up to BCNF).This means give a new schema (either as an ER diagram or SQL CREATE TABLE statements) that is a normalised version of the original.

Listing 1: question1

```
drop database if exists student;
create database student;
use student;

drop table if exists 'grade';
drop table if exists 'stuId';

create table 'stuId'(
    'stuId' varchar(100) not null primary key,
    'name' varchar(100) not null,
    'gender' varchar(1) not null
);

create table 'grade'(
    'stuId' varchar(100) not null primary key ,
    'unit' varchar(100) not null,
    'grade' INTEGER not null ,
    CONSTRAINT copop_city foreign key ('stuId') references stuId('stuId')
);
```

Question 2

The CIA world factbook contains geographical, political and military information about the world. Here is part of one table listing principal cities from 2015:

We will assume for this exercise that city names are globally unique and therefore the "City" column has been chosen as the primary key for this table. The "pop" column lists the city's population and the "co_pop" lists the population of the country in which the city is located (with abbreviations K = 1000, M=1000000). The "capital" column is a Boolean yes/no value that is set to "yes" for exactly one city in each country. (While the capital is included in the table for every country however small, non-captial cities are only included if they are of international

Question 2 continued on next page...

city	country	pop	co_pop	capital
Paris	France	10.843M	66.8M	yes
Lyon	France	1.609M	66.8M	no
Marseille	France	1.605M	66.8M	no
Papeete	French Polynesia	133K	285K	60
Libreville	Gabon	707K	1.7M	60

significance.)

(a) Identify the candidate key(s) in every table.

$$Relation = \{city, country, pop, co\_pop, capital\}$$

$$Function = \begin{Bmatrix} city \rightarrow country, \\ city \rightarrow pop, \\ country \rightarrow co\_pop, \\ city \rightarrow captial \end{Bmatrix}$$

In this way, we can get closure city:

$$\begin{aligned} (city)^+ &= \{city\} \\ &= \{city, country\} \quad (city \rightarrow country) \\ &= \{city, country, pop\} \quad (city \rightarrow pop) \\ &= \{city, country, pop, co\_pop\} \quad (country \rightarrow co\_pop) \\ &= \{city, country, pop, co\_pop, captial\} \quad (city \rightarrow captial) \end{aligned}$$

$$So\ we\ get : (city)^+ \sqsupseteq Relation$$

So city is a candidate key.

(b) Identify the key and non-key attributes in every table.

Because city is the only candidate key we get, so city is the candidate key and primary key. Country, pop, co_pop, captial is non-key.

(c) Determine which normal forms from (1NF, 2NF, 3NF, BCNF) the schema does or does not satisfy. Give evidence to support your answer.

For function in this table:

$$Let:\ Function = F\_city \cap F\_country$$

$$F\_city = \begin{Bmatrix} city \rightarrow country, \\ city \rightarrow pop, \\ city \rightarrow captial \end{Bmatrix}$$

City is candidate key, so F_city is satisfy in BCNF. In F_country from Function, co_pop is

non-key, and country is not a candidate key for Function but a candidate key for F_country.

$$F\_country = \{country \rightarrow co\_pop\}$$

In this way, both F_city and F_country are satisfy in BCNF, but Function is not satisfy in BCNF.
Function is just satisfy in 2NF, for both country and co_pop are non-key .

(d) If the schema is not in BCNF, normalise it as far as possible(up to BCNF).This means give a new schema (either as an ER diagram or SQL CREATE TABLE statements) that is a normalised version of the original.

Listing 2: question2

```
drop database if exists cities_2015;
create database cities_2015;
use cities_2015;

drop table if exists `co_pop`;
drop table if exists `city`;

create table `city`(
    `city` varchar(100) not null primary key,
    `country` varchar(100) not null,
    `pop` INTEGER not null,
    `capital` BOOLEAN not null
);

create table `co_pop`(
    `city` varchar(100) not null primary key ,
    `co_pop` INTEGER not null,
    CONSTRAINT copop_city foreign key (`city`) references city(`city`)
);
```