





DATA SCHEMA

TABLE CREATING QUARIES

```
-- table of genes
DROP TABLE IF EXISTS cs275_gene;
CREATE TABLE cs275_gene(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    chromosome VARCHAR(255) NOT NULL,
    coordinates VARCHAR(255) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY (name, chromosome, coordinates)
) ENGINE=InnoDB;

-- table of patients
DROP TABLE IF EXISTS cs275_patients;
CREATE TABLE cs275_patients(
    id INT NOT NULL AUTO_INCREMENT,
    fname VARCHAR(255) NOT NULL,
    lname VARCHAR(255) NOT NULL,
    lid INT(11),
    PRIMARY KEY (id),
    UNIQUE KEY (fname, lname, lid),
    FOREIGN KEY (lid) REFERENCES cs275_gene(id) ON DELETE SET NULL
) ENGINE=InnoDB;

-- table of mutations
DROP TABLE IF EXISTS cs275_mutation;
CREATE TABLE cs275_mutation(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    notes TEXT,
    PRIMARY KEY (id),
    UNIQUE KEY (name)
) ENGINE=InnoDB;

-- table of primers
DROP TABLE IF EXISTS cs275_primer;
CREATE TABLE cs275_primer(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    lid INT(11),
    PRIMARY KEY (id),
    UNIQUE KEY (name),
    sequence VARCHAR(id);
    direction VARCHAR(id);
    FOREIGN KEY (lid) REFERENCES cs275_gene(id) ON DELETE SET NULL
) ENGINE=InnoDB;
```

```

--reverse primer
DROP TABLE IF EXISTS cs275_primer_reverse;
CREATE TABLE cs275_primer_reverse(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    lid INT(11),
    sequence VARCHAR(id);
    direction VARCHAR(id);
    PRIMARY KEY (id),
    UNIQUE KEY (name),
    FOREIGN KEY (lid) REFERENCES cs275_gene(id) ON DELETE SET NULL
) ENGINE=InnoDB;

```

```

--premir pair
DROP TABLE IF EXISTS cs275_primer_pair;
CREATE TABLE cs275_primer_pair(
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    ABV DECIMAL(3,1),
    sid INT(11),
    bid INT(11),
    cid INT(11),
    description TEXT,
    PRIMARY KEY (id),
    UNIQUE KEY (name),
    FOREIGN KEY (sid) REFERENCES cs275_mutation(id) ON DELETE SET
NULL,
    FOREIGN KEY (bid) REFERENCES cs275_primer(id) ON DELETE SET NULL
    FOREIGN KEY (cid) REFERENCES cs275_primer_reverse(id) ON DELETE
SET NULL
) ENGINE=InnoDB;

```

```

-- table for selection primers that worked the best for the patients
DROP TABLE IF EXISTS cs275_best_combination;
CREATE TABLE cs275_best_combination(
    uid INT(11),
    bid INT(11),
    PRIMARY KEY (uid, bid),
    FOREIGN KEY (uid) REFERENCES cs275_patients(id) ON DELETE
CASCADE,
    FOREIGN KEY (bid) REFERENCES cs275_primer_pair(id) ON DELETE
CASCADE
    FOREIGN KEY (cid) REFERENCES cs275_primer_pair(id) ON DELETE
CASCADE
) ENGINE=InnoDB;

```

INSERT QUERIES

```
-- Project: Insert Data

-- insert patient
INSERT INTO cs275_patient (fname, lname, lid)
VALUES([fname], [lname], (
SELECT id FROM cs275_gene
WHERE name = [name] AND chromosome = [chromosome] AND coordinates =
[coordinates]))
ON DUPLICATE KEY UPDATE fname = VALUES(fname),
lname = VALUES(lname),
lid = VALUES(lid);

-- insert primer_pair
INSERT INTO cs275_primer_pair (name, ABV, sid, bid, cid description)
VALUES([name], [ABV],
(SELECT id FROM cs275_mutation WHERE name = [name]),
(SELECT id FROM cs275_primer WHERE name = [name]), [description])
ON DUPLICATE KEY UPDATE name = VALUES(name),
ABV = VALUES(ABV),
sid = VALUES(sid),
bid = VALUES(bid),
cid = VALUES(cid),
description = VALUES(description);

-- insert primer
INSERT INTO cs275_primer (name, lid, sequence, direction)
VALUES([name], (
SELECT id FROM cs275_gene
WHERE name= [name] AND chromosome = [chromosome] AND coordinates =
[coordinates]))
ON DUPLICATE KEY UPDATE name = VALUES(name),
lid = VALUES(lid);

-- insert primer reverse
INSERT INTO cs275_primer_reverse (name, lid, sequence, direction)
VALUES([name], (
SELECT id FROM cs275_gene
WHERE name= [name] AND chromosome = [chromosome] AND coordinates =
[coordinates]))
ON DUPLICATE KEY UPDATE name = VALUES(name),
lid = VALUES(lid);

-- insert mutation
INSERT INTO cs275_mutation (name, notes)
VALUES ([name], [notes])
ON DUPLICATE KEY UPDATE name = VALUES(name),
notes = VALUES(notes);
```

```

-- insert gene
INSERT INTO cs275_gene(name, chromosome, coordinates)
VALUES ([name],[chromosome],[coordinates])
ON DUPLICATE KEY UPDATE name = VALUES(name),
chromosome = VALUES(chromosome),
coordinates = VALUES(coordinates)

-- insert favorite primer_set
INSERT INTO cs275_best_combination(uid, bid)
VALUES(
(SELECT id FROM cs275_users WHERE fname = [fname] AND lname =
[lname]),
(SELECT id FROM cs275_primer_pair WHERE name = [name]))
ON DUPLICATE KEY UPDATE uid = VALUES(uid),
bid = VALUES(bid), cid = VALUES(cid);;

```

DELETE QUERIES

-- Project: Delete Data

-- delete patients

```

DELETE FROM cs275_patients
WHERE id = [patients id];

```

-- delete primer sets

```

DELETE FROM cs275_primer_pair
WHERE id = [];

```

-- delete primer

```

DELETE FROM cs275_primer
WHERE id = [primer id];

```

-- delete primer

```

DELETE FROM cs275_primer_reverse
WHERE id = [primer_reverse id];

```

-- delete mutation

```

DELETE FROM cs275_mutation
WHERE id = [mutation id];

```

-- delete gene

```

DELETE FROM cs275_gene
WHERE id = [location gene];

```

```
-- delete good primers
DELETE FROM cs275_best_combination
WHERE uid = [patient id] AND bid = [primer_pair id];
```

SELECT QUERIES

```
-- Project: Select Data
--selection patients
SELECT U.fname, U.lname, L.name, L.chromosome, L.coordinates
FROM cs275_patients U
LEFT JOIN cs275_gene L ON L.id = U.lid
ORDER BY U.lname, U.fname ASC)))
```

```
--select a primer_set
SELECT BE.name, BE.ABV, S.name, BR.name, BR.sequence, R.name, R.sequence, BE.description
FROM cs275_primer_pair BE
LEFT JOIN cs275_mutation S ON S.id = BE.sid
LEFT JOIN cs275_primer BR ON BR.id = BE.bid
LEFT JOIN cs275_primer_reverse R ON R.id = BE.cid
ORDER BY BE.name ASC"
```

```
--select a primer
"SELECT BR.name, L.name, L.chromosome, L.coordinates, BR.sequence, BR.direction
FROM cs275_primer BR
LEFT JOIN cs275_gene L ON L.id = BR.lid
ORDER BY BR.name ASC"
```

```
--select a primer_reverse
"SELECT BR.name, L.name, L.chromosome, L.coordinates, BR.sequence, BR.direction
FROM cs275_primer BR
LEFT JOIN cs275_gene L ON L.id = BR.lid
ORDER BY BR.name ASC"
```

```
--mulect mutation
SELECT name, chromosome, coordinates
```

```
FROM cs275_gene  
ORDER BY chromosome ASC
```

```
--selection a primer combination that worked for the patients  
SELECT U.fname, U.lname, BE.name, BR.name, R.name, R.sequence, BR.sequence  
FROM cs275_patients U  
INNER JOIN cs275_best_combination FB ON FB.uid = U.id  
INNER JOIN cs275_primer_pair BE ON BE.id = FB.bid  
LEFT JOIN cs275_primer BR ON BR.id = BE.bid  
LEFT JOIN cs275_primer_reverse R ON R.id = BE.cid  
ORDER BY U.lname, U.fname ASC"
```

UPDATE QUERIES

I HAVE NOT FINISHED THE UPDATE SECTION OF THE PROGRAM
AT THIS POINT ONLY CERTAIN VARIABLES CAN BE UPDATED

```
-- Project: Update Data
```

```
-- patients  
SELECT id, fname, lname  
FROM cs275_patients  
ORDER BY lname ASC
```

```
--genes  
SELECT id, name, chromosome, coordinates  
FROM cs275_gene  
ORDER BY chromosome ASC
```

```
--primer pair  
SELECT id, name  
FROM cs275_primer_pair  
ORDER BY name ASC
```

```
--select primer
```

```
SELECT id, name, sequence, direction
```



```
FROM cs275_primer
ORDER BY name ASC
```

```
-- reverse primer
SELECT id, name, sequence, direction
FROM cs275_primer
ORDER BY name ASC
```

```
--select mutation
SELECT id, name
FROM cs275_mutation
ORDER BY name ASC
```

Outline of Database

The database is multifunctional. It allows to keep track of people who tested for specific mutation and figure out which primer sets are the good ones and which patient can be used as a positive control because the patient produced good results. At the same time it will allow us to keep track of the primer inventory for each gene. Another feature of the database is that, new primer sets can be created based on the information of the genomic coordinates and information if the primer is F or R. At the same time, we will be able to keep track of which patient was tested with which primer set.

How the Database Works

There are 7 entity tables: **cs275_gene(gene)**, **cs275_mutation (mutation)**, **cs275_primer(primer)**, **cs275_primer_reverse (reverse)**, **cs275_primer_pair (pair)** and **cs275_patients (patients)** and **ce_275_best_combination(best)**. Each of these entity tables contains at least two attributes in addition to a primary key. For each table, the primary key is a single integer that is not null and auto-increments with each new row. The **best** table is slightly different because it does not have an auto-incrementing integer. Instead it has a combined primary key of its two foreign key references.

The **patient** table contains an integer **id** attribute as the primary key. Other **patient** attributes include **fname**, **lname** and **lid**. The unique key is a combination of **fname**, **lname** and **lid**. The varchar **fname** and **lname** represent the patient's first and last names respectively, and these attributes cannot be null. The integer **lid** is a foreign key that references the primary key of the **gene** table. The **patient** table is related to the **gene**, **best** and **primer_set** by a many-to-many relationship. A patient can be tested for different genes' mutations and it can be found that for different genes there is a different best primer set and a patient can be tested using

different primer sets. At the same time, best primer set can be best for multiple patients and multiple primer sets can be used to test patient as well as genes of multiple people will be tested.

The **primer_set** table contains an integer **id** attribute as the primary key. The **primer_pair** has **name**, **temperatureABV**, **sid**, **bid**, **cid**, and **description**. The unique key is the **name**. The **name** attribute is a varchar that represents the name of the primer_set, and cannot be null. **ABV** is a decimal representing melting temperature of the PRC reaction, and **description** is a text to indicate any useful information about the PCR conditions. The integer **sid** is a foreign key that references the primary key of the **mutation** table, and the integer **bid** attribute is a foreign key that references the primary key of the **primer**, while the integer **cid** is a foreign key that reference the primary key of the **reverse_primer**. The **mutation** table is related to **primer_set** as a M to M relationship. Primer_set can detect different mutations and different mutations can be detected by different primer sets. Primer_set table has a M to M relationship with the patient table because patient can be tested with different primer set and at the same time primer set can be used to test different patients.

The **primer** and **primer_reverse** table contains an integer **id** attribute as the primary key. Other attributes for the **primer** and **primer_reverse** table include **name** and **lid**. The unique key is the **name**. **name** is a varchar representing the name of the primer. The **lid** attribute is a foreign key that references the primary key of the **gene** table. The **primer** and **primer_revers** table is related to the **gene** by a one-to-many relationship and the **location** table by a many-to-one relationship. Different primers can be used to amplify the gene, but the primer amplifies only one gene. The **mutations** table contains an integer **id** attribute as the primary key. Also included in the **styles** table are **name** and **notes** attributes. The unique key is the **name**. As with **primer_pair**, **primer** and **primer_reverse**, **name** is a varchar representing the name of the style. **note** is a text attribute for keeping notes on style types. The **gene** table contains an integer **id** attribute as the primary key. Locations also includes varchar **name**, chromosome and genomic coordinates.

Tatyana Vlaskin

References will be provided upon request.