Assignment 01

Vivek Rai, 12BT30025

August 9, 2015

1 SQL Queries

1.1 Ans

```
SELECT product.pid, product.name
FROM product, inventory
WHERE product.pid = inventory.pid AND inventory.stock < 5;
```

1.2 Ans

```
SELECT DISTINCT supplier.sid, supplier.name
FROM supplies
JOIN manufactures USING(pid)
JOIN supplier USING(sid)
JOIN manufacturer USING(mid)
WHERE manufacture.name = 'manufacturer_2';
```

1.3 Ans

```
SELECT product.pid, name, inventory.stock
FROM product, inventory
WHERE product.pid = inventory.pid
ORDER BY stock DESC;
```

1.4 Ans

```
SELECT pid, name, COUNT(*) as `num`
FROM product
JOIN supplies USING(pid)
GROUP BY pid HAVING num = 1;
```

1.5 Ans

```
SELECT pid, product.name
FROM product
JOIN inventory USING(pid)
WHERE stock = (SELECT MIN(stock) FROM inventory);
```

1.6 Ans

```
SELECT sid, name, COUNT(*) as `product_supplied`
FROM supplier
JOIN supplies USING (sid)
GROUP BY sid;
```

1.7 Ans

```
SELECT manufactures.mid, manufacturer.name, AVG(min_age)
FROM manufactures
JOIN product USING(pid)
JOIN manufacturer USING(mid)
GROUP BY mid
ORDER BY min_age DESC
LIMIT 1;
```

1.8 Ans

Which manufacturer has highest throughout?

```
SELECT mid, name, COUNT(*) AS `products_produced`
FROM manufacturer
JOIN manufactures USING(mid)
GROUP BY mid
ORDER BY `products_produced` DESC
LIMIT 1;
```

2 Integrity Constraint

2.1 Ans

After specifying primary keys for the tables, adding further values requires the value corresponding to primary key to be unique, else the operation is rejected.

For example, if we try to insert into product table, a new tuple with pre-existing pid but new name or min_age, it will be rejected. Without this constraint,however, it will be added perfectly to the table.

2.2 Ans

MySQL database schema when used with foreign keys, allows *referential integrity*. It provides a flexible way of reducing data duplication, prevent changes from happening or propagating changes across different tables with minimum action.

In case we specify a foreign key relation between two tables, and try to perform a **INSERT** or **UPDATE** operation on the child table corresponding to which there is no entry in parent table, the operation is rejected. For example,

the situation of inserting a product in inventory table corresponding to which no entry exists in Product table.

This obviously depends on the parent-child relation and the way foreign-key constraints are specified. In case a match is found, further behavior is controlled by a different set of constraints, **CASCADE**, **RESTRICT**, for example.

Not having these constraints will allow different inconvenient situations where we have data duplication, manual effort of adding/updating data to each table and ensuring that they are correct etc.,