Lab : 14th November 2014

# **MySQL recovery using GUI**

For this lab , I worked with the world database.

In the sheet I was told to do two different types of back up (One backing up all tables, another backing up just a single table). "Export to Dump Project Folder" and "Export to Self-Contained File".

I first did a full Dump of the database , using the Gui interfaces.

I then dropped the database using the following sql command

* DROP DATABASE world;

I then checked to see if the database was useable .

* Use world;

I got an error saying it couldn't be used.

I then used the import facility in Workbench to import the backup. I then ran **SELECT \* FROM city** to make sure everything was restored.

I then used the second export option to backup a single table.

I backed up the table called city.

I then deleted this table and imported the backup to restore it.

I ran **SELECT \* FROM city** to make sure it was restored, just in case.

The naming conventions of the backups were as follows: **Dump20141121**.

It uses the year, month and day to create the name. This is important as if you were making different backups week and at some point needed to use one of these backups and would be able to choose the relevant one.

Ingres Backup and Restore

For this lab we looked at back up and restore but in ingures. it was harder then sql.

Section 1: Basic Backup

Create database 'recovery'

CREATEDB recovery;

Create table 'orders'

CREATE TABLE orders (

order\_no Integer,

issued timestamp

);

Create Backup

Then created a backup of the database using the command ckpdb recovery.

Begin transaction/insert two rows into 'orders'

BEGIN TRANSACTION

INSERT INTO orders

VALUES (1, CURRENT\_TIMESTAMP);

INSERT INTO orders

VALUES (2, CURRENT\_TIMESTAMP);

commit;

5. Backup the database again and restoring

I first deleted the first backup I had made with the following command: ckpdb recovery.

After this I created another backup.

I then displayed the rows in the table.

SELECT \*

FROM orders

This returned the two rows which I previously entered.

I then deleted the two rows from the table and attempted to display them again.

They did not appear as they have been deleted.

I then restored the database with the backup.

* rollforwarddb recovery.

I then ran SELECT \* FROM orders and the two rows were restored.

Section 2: Backup and Logs for Point in Time Recovery

First I had to enable journaling for the database. I did this by creating a backup of the databse. i used the follwoing command

* ckpdb +j recovery.

To make use it was working/running .

I ran

* Infodb recovery

I then inserted a third row into the database

BEGIN TRANSACTION

INSERT INTO orders

VALUES(3, CURRENT\_TIMESTAMP)

commit;

I then viewed all rows in the table.

I then created another transaction.

BEGIN TRANSACTION

SELECT \* FROM orders

DELETE FROM orders

SELECT \* FROM orders

commit;

The table now has no data in it.

I then ran the auditdb recovery command to find the transaction times.

* I found the end time for the insert row : 21-Nov-2014 11:41:10.01
* I then found the begin time for the delete transaction : 21-Nov-2014 11:44:54.50

I then went and restored the databse to before the start of the delete transction

rollforwarddb +j -e21-Nove-2014:11:41:00 recovery

After this I ran sql below to check it worked.

SELECT \*

FROM orders

The three rows had been restored.