# CS344: Class Notes

**Topic:-Review Session (Continued)**

**Date: - Sep 13 2011**

**Prepared by:-**

**Raman Anurag** **09010139**

**Ravi Sethia** **09010140**

**Table operations:**

(Learn syntax from previous lectures)

Create table – each table must have one primary key.

Delete, drop, select, average, min, max etc. are some other queries.

Group by – when any aggregate function is used then those attributes should be present in group by.

Nested SQL queries

**Normal forms:**

Need of normal form –

- Remove redundancy
- Multiple searches
- Evaluating quality of ER model
- Updating data
- Deletion of anomalies

**1 NF** – atomic domain

**2 NF** – if $\alpha \rightarrow \beta$ ($\alpha$ derives $\beta$) then no proper subset of $\alpha$ should be part of a candidate key.

**BCNF** – if $\alpha \rightarrow \beta$ then either it is trivial that is $\beta$ is a subset of $\alpha$ or $\alpha$ is the primary key of the table.

**3 NF** - if $\alpha \rightarrow \beta$ then either it is trivial that is $\beta \subseteq \alpha$ or $\alpha$ is the primary key or $\beta - \alpha$ should part of a candidate key.

**Advantages of Normal forms:**

1. Guarantees functional dependencies
2. If we change one value then no need to change multiple rows
3. Smaller tables
4. Entities properly separated
5. Dependencies only from keys

Note: In BCNF dependency may not be preserved.

If we brake a table R into R1 and R2 such that R1 $\cup$ R2 = R

If R, R1 and R2 have functional dependencies F, F1 and F2 respectively, then dependency is preserved if F+ = (F1 U F2) +

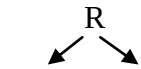
**Revise Armstrong's rules.**

Decomposition:

Convert functional dependencies (F) to minimal cover (M).

If $\alpha \rightarrow \beta$ and $|\beta| = 1$ then F+ = M+.

Algorithm for converting into normal forms:




R1 R2 $R1 \cap R2 = \alpha$

$(R - \beta)$ $(\alpha - \beta)$

$\alpha$ is the primary key for R2 so it is lossless decomposition. R2 is in 2-NF. Similarly prove R1 is in 2 - NF, if it is not, decompose further.

Prove termination: As no of attributes after decomposition reducing.

Same algorithm for BCNF and 3 – NF, just change termination condition.

More notes:

1. By decomposition in BCNF we may lose dependency
2. In 3 NF dependencies are preserved
3. All BCNF are also 3 NF

- Structured data - ER, Relational
- Semi Structured – XML (for portability across machines)
- Unstructured – text

Since this was a review session most things have already been covered in previous notes refer to them for details .We have highlighted the important points discussed and overview of topics covered

**Revise inter-conversions and tree structure.**