#### CSCI 4140 – Master Notes

Contents

[CSCI 4140 – Master Notes 1](#_Toc133934397)

[Week 1: 1](#_Toc133934398)

[List of Topics: 1](#_Toc133934399)

[Relational Data Models and Relational Algebra: 1](#_Toc133934400)

[Evolution of Relational DB Systems 1](#_Toc133934401)

[Keys: 1](#_Toc133934402)

## Week 1:

### List of Topics:

* Relational Data Model and Relational Algebra
* DB Modeling
* SQL
* NoSQL – MongoDB
* Transaction Management
* 2-Phase Commit
* DB life-cycle and Performance
* Additional (???)

### Relational Data Models and Relational Algebra:

1.1 Relational Data Model and Properties – History, Model, Keys, Indices, Constraints  
1.2 Relational Algebra – Basic Relational Operators, Joins, Divisions

### Evolution of Relational DB Systems

* Prior to 1960s: File systems accessed by independent programs
* 1960s: Computerized database stared in 60s, two popular data models
* Network model called CODASYL
* Hierarchical model called IMS
* 1970 to 1972: E.F. Codd published a relational database model in which the databases schema (logical organization) was disconnected from physical information storage… simple yet defined on math (relational algebra) – business and academia liked it
* Mid 1970s: Two major relational database system prototypes were created between the years 1974 and 1977:
* Ingres, which was developed at UBC, used QUEL language - MS SQL Server, Sybase
* System R, created by IBM San Jose- used SEQUEL language e.g. DB2, Oracle
* 1976: P. Chen published Entity Relationship (ER) for Data Modelling
* 1980s: Structured Query Language (SQL) became the standard query language
* Relational Data Model – by CODD IN 1970
* Was embraced by industry and academia (R&D)
* Simple, widely understood
* Based on Set Theory – rigorous and hence amenable to research and development in a logical and rigorous manner

### Keys:

* Designer determines based on application semantics
* Semantics are derived by talking to users
* Composite key vs single-column key
* Key attribute
* Superkey (or simply a key)
* Candidate Key
* Irreducible attributes to ensure uniqueness (cannot remove any attribute)
* Primary key – DB chosen candidate key
* Null: absence of any data value
* Entity Integrity Constraint (unique, not null)
* Foreign Key Constraint
* Secondary Key – DB concept only
* Not in Relational Data Model as secondary keys may not be unique (e.g. name used for secondary key for efficient search)

**Several different types of keys are used in the relational data model:**

* Composite Key: key that is composed of more than one attribute
* Key Attribute: attribute that is part of a key
* Superkey: key that can uniquely identify any row in the table
* (Missing – Review Notes)

|  |  |
| --- | --- |
| **Table 3.1** | **Characteristics of a Relational Table** |
| 1 | A table is perceived as a two-dimensional structure composed of rows and columns. |
| 2 | Each table row (tuple) represents a single entity occurrence within the entity set. |
| 3 | Each table column represents an attribute, and each column has a distinct name. |
| 4 | Each intersection of a row and column represents a single data value. |
| 5 | All values in a column must conform to the same data format (all values must be from a domain set) |
| 6 | Each column has a specific range of values known as the attribute domain. |
| 7 | The order of the rows and columns is immaterial to the DBMS. |
| 8 | Each table must have an attribute or combination of attributes that uniquely identifies each row (each table must have a primary key). |