AJ Ricketts

Prof Mengchi Liu

Saturday October 19th, 2019

COMP 3005 Assignment #3

**Concepts**

1. **Mini Data World:** Mini Data world refers to conceptual data models, these data models provide concepts that are close to the way many perceive data. This is a high level data model.
2. **Data Model:** A data model specifies how data is structured and operated. It is comprised of three parts, a set of concepts that describe the structure of the database, a set of operations to manipulate these structures, and a set of constraints that the database should obey.
3. **Database System:** There are two types of database management systems, centralized, and client-server. Centralized combines everything into a single system (DBMS software, hardware, etc), and client-server has the DBMS and the database as the client, and then an application for the user to use as the client.
4. **Domain:** Domains represent the grouping of values together.
5. **Relational Model:** In the relational model, all data is represented in terms of tuples, and grouped into relations.
6. **Attribute:** Attributes are the groupings the values in a database are divided into.
7. **Relation:** A relation is the grouping of the attributes and their values. These attributes are all related in some way. For example the student relation from the slides.
8. **Primary Key:** Primary Key is a chosen key in a relation that is used to uniquely identify each tuple in a relation. It can also be used to reference the tuple from another tuple. The primary key attribute is underlined when displayed (when drawing out a relation or on the slides).
9. **Logical Data Independence:** The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.
10. **SQL:** SQL is a popular standardized query language used for managing data held in a relational database management system (RDBMS)
11. { N, HN | (exists P#) ( Person( P#, N, \_) and N = ‘Lastname’ and (exists H#) Hobby(H#, HN) and Play( P#, H#, \_))}

**Result: Name**

Chess

Dancing

1. { N| (exists P#, H#) ( Person( P#, H#, \_) and Hobby( H#, ‘Bowling’) and Play (P#, H#, \_))}

**Result: Name**

Smith

Jones

1. { N| (exists P#, H#, T) (Person( P#, N, \_) and Play( P#, H#, T > 3))}

**Result: Name**

Smith

Jones

Lastname

1. { N| (exists P#, H1#, H2#) (Person( P#, N, \_) and (( Play( P#, H1#, \_) and Hobby(H1#, ‘Chess’)) or (Play( P#, H2#, \_) and Hobby( H2#, ‘Dancing’))))}

**Result: Name**

Smith

Jones

Blake

Lastname

1. { N| (exists P#, H1#, H2#) (Person( P#, N, \_) and (( Play( P#, H1#, \_) and Hobby(H1#, ‘Chess’)) and (Play( P#, H2#, \_) and Hobby( H2#, ‘Dancing’))))}

**Result: Name**

Smith

Jones

Blake

Lastname

1. { N, HN | (exists P#, H#) (Person( P#, N, \_) and Hobby(H#, HN) and Play(P#, H#, \_))}

**Result: Name Name**

Smith Bowling

Smith Chess

Smith Dancing

Smith Hiking

Smith Skating

Smith Ski

Jones Bowling

Jones Chess

Jones Dancing

Jones Hiking

Blake Chess

Blake Dancing

Lastname Chess

Lastname Dancing

1. { N | (exists P#) (Person (P#, N, \_) and (forall H#) (if (exists HN) Hobby( H#, HN) and HN != ‘Ski’) then Play( P#, H#, \_))}

**Result: Name**

Jones

Blake

Lastname

Adam

1. { N | (exists P#) (Person (P#, N, \_) and not (exists H#) Play( P#, H#, \_))}

**Result: Name**

Adams

1. { N | (exists P#) Person( P#, N, \_) and (forall H#) not Hobby( H#, \_) or Play(P#, H#)}

**Result: Name**

Smith

1. { N | (exists P#’) (Person( P#’, N, \_) and N != ‘Lastname’ (exists P#) Person(P#, ‘Lastname’, \_ ) and (forall H#) ( if ( Hobby(H#, \_) and Play( P#, H#, \_) then Play( P#’, H#, \_ ))))}

**Result: Name**

Smith

Jones

Blake

1. { N | (exists P#’) (Person( P#’, N, \_) and N != ‘Lastname’ (exists P#) Person(P#, ‘Lastname’, \_ ) and (forall H#) ( if ( Hobby(H#, \_) and Play( P#, H#, \_) then Play( P#’, H#, \_ ) or ( if ( Hobby(H#, \_) and not Play( P#, H#, \_) then not Play( P#’, H#, \_ ))))}

**Result: Name**

Blake

1. { N | (exists P#) (Person( P#, N, \_) and (forall H#) ( if (exists HN) (Hobby( H#, HN) and HN != ‘Skating’ and HN != ‘Ski) then Play (P#, H#) or ( if (exists HN) (Hobby( H#, HN) and HN = ‘Skating’ and HN = ‘Ski) then not Play (P#, H#))}

**Result: Name**

Jones



**Name Count Total**

-----------------------------------------------------

Smith 6 13

Jones 4 14

Blake 2 05

Lastname 2 07

**Result: Name**

Blake

Lastname