Relationships between Tables and Entity Relationship Diagrams

FIT 100

FIT 100 Remember Database Terminology…

- A database is a set of tables
- The structure of a database is displayed in its schema
- A database instance is the current contents of a database

Attribute (Field)(Column)

Table Name (Entity)

Top_Scholar

Tuple (Row) (Record)

Student_ID	Nick_Name	Major	GPA
1021253	Chela	INFO	3.89
1021343	J.T.	SPAN	3.85

- Fields (columns) have a type …all items in a column have that type
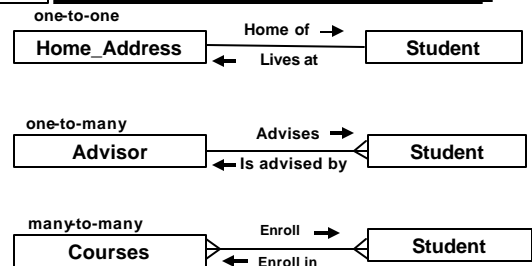- A database table stores information about entities

---

FIT 100 Logical Relationships Among Tables

- A database schema will represent the structure of your database. This includes the distinguishing attribute(s) of every entity- the Primary Key
- Keys are also the way to represent relationships between entities
- The diagram (model) that shows how tables are related is known as the Entity Relationship Diagram (ERD)
- Separating information out into separate tables and establishing relationships among those tables allows database designers to avoid redundancy and keep data accurate
- The student information database shown in class last week represented a one-to-one relationship
- There are also one-to-many relationships and many-to-many relationships. You have seen one example in Lab.

FIT 100 Sample Entity Relationship Diagrams

one-to-one

| Home_Address | Home of → | Student |
| | ← Lives at | |

one-to-many

| Advisor | Advises → | Student |
| | ← Is advised by | |

many-to-many

| Courses | Enroll → | Student |
| | ← Enroll in | |

Relationships between tables will determine the key structures of the tables

How to Structure Tables to Show Relationships

FIT 100

❖ In a one-to-one relationship, tables are associated together through the primary key in each.

❖ In a one-to-many relationship, the primary key attribute in the one must be listed as an additional attribute in the many. The tables are associated together through the similar attributes.

❖ In a many-to-many relationship, a new table is ALWAYS created and the primary key attributes of the original tables are made attributes of the new "bridge" table and are often combined to form the primary key – or a surrogate key is created

---

Operations on Databases

FIT 100

Tables are useful, but they become much more powerful when we can manipulate them to create new tables from existing tables. For that, we need special operations.

Tables that Produce Other Tables

FIT 100

❖ Table operations can involve one or many tables

❖ These basic operations are usually used together to create specific "views" of the database

❖ Let's look at the basic operations performed on tables…

Basic Operations On Tables

FIT 100

❖ An obvious use of tables would be to construct other tables from them.

❖ The Dean's list of students shown last week was a table constructed from other tables using several of the following operations: Select, Project, Union, Difference, Product

Dean's View of Database

Student_ID	Nick_Name	Major	GPA	Street_Address	City	State	Country	PostalCode
1021253	Chela	INFO	3.89	14 Mountain Ave	Victoria	BC	Canada	V6N4T4
1021343	J.T.	SPAN	3.85	1715 65th Ave	Seattle	WA	USA	98125
…	…	…	…	…	…	…	…	…

❖ This table does exist by itself. It is a view of certain rows and columns from other tables.

A Sample Schema

Advisor
      Advisor_ID   Integer
      FirstName   String
      LastName   String
      Department String
      HireDate   Date

PK: Advisor_ID

Table Operation: Select

❖ Select picks a subset of records (rows) based on some criteria
    Select_from \<table\> **On** \<T/F condition\>

❖ We could create a subset from the Advisors table of just those advisors from the CSE Department

Select_from Advisors **On** Department = "CSE"

Advisor_ID	FirstName	LastName	Department	HireDate
1	Larry	Snyder	CSE	1988
5	Alan	Borning	CSE	1992
24	Jeremy	Baer	CSE	1998

CSE_Advisors=**Select_from** Advisors **On** Department = "CSE"

Table Operation: Project

❖ Project extracts columns from a table
    Project \<attribute list\> **From** \<table\>
❖ Picking out the essential information from CSE_Advisor
    Project FirstName, LastName **From** CSE_Advisor

Advisor ID	FirstName	LastName	Department	HireDate
1	Larry	Snyder	CSE	1988
5	Alan	Borning	CSE	**CSE_Advisor**
24	FirstName	LastName	CSE	1998
	Larry	Snyder		
	Alan	Borning	**Name_CSE_Advisor**	
	Jeremy	Baer		

NameCSE_Advisor = **Project** FirstName , LastName **From** CSE_Advisor

---

Table Operations: Union and Difference

❖ Union combines two tables with like attributes
    \<table\> **+** \<table\>

❖ Example: You want to combine two tables of scheduled games for different teams. One shows home games, the other away games. All the attributes are the same.

❖ Difference removes a table from a table with like attributes
    \<table\> **-** \<table\>

❖ Example: You want to remove all games starting after 5PM from your schedule of games

FIT 100 Table Operation: Product
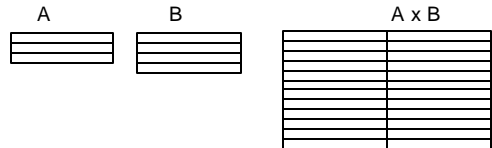
- Product multiplies two tables together creating a "super table"

- For each row in the first table, concatenate every row in the second table

  <table> **x** <table>

- Product can be used to add new fields

- Product's Column Rule: If TableA has m columns and TableB has n columns, then TableA_x_TableB has m+n columns

- Product creates a table of "all pairs"

- Product's Row Rule: If TableA has m rows and TableB has n rows, then TableA_x_TableB has mn rows

---

FIT 100 Product: The Rules Always Apply

- Visualize a Product …

  A   B   A x B

- The row and column rules always apply

- Column Rule: If TableA has m columns and TableB has n columns, then TableA_x_TableB has m+n columns

- Row Rule: If TableA has m rows and TableB has n rows, then TableA_x_TableB has mn rows

---

FIT 100 Join – Product With a Match

- Join is a combination performing a product and a select

- The Join symbol in database research literature is a "bow tie"
  $\bowtie$

- Natural Join… suppose two tables have the same attribute, then use the Product operation to pair all rows of the two tables, but keep only those rows that match on the common attribute and remove duplicates

- Other joins are those done with other relational operators:
  <, >, <=, etc.

FIT 100 Constructing the Dean's View

- The Dean's View, which doesn't literally exist, is created by joining two tables

Home_Base   Top_Scholar
    Student_ID   Student_ID
    Street_Address   Major
    Nick_Name   GPA
    City
    State   PK: Student_ID
    Country
    PostalCode
PK: Student_ID

- The common field is student ID, so…
  Top_Scholar $\bowtie$ Home_Base creates the result