

Chapter 3: Specifics on The Relational Model

CSCI 475

General Relational Model Terminology

Database → Relations → Tuples → Attributes

Employee Table:

emp#	Name	login	HourlyRate	NumDependents
3	Bob	bsmith	12.50	1
9	Carol	cjones	24.00	2
5	Ted	twilson	19.50	3

Degree - # of fields, ex. degree = 5

Cardinality - # of tuples/records, ex. cardinality = 3

Field Data types – numeric, character/text, date, logical

General Relational Model Terminology

Database → Relations → Tuples → Attributes

Employee Table:

emp#	Name	login	HourlyRate	NumDependents
3	Bob	bsmith	12.50	1
9	Carol	cjones	24.00	2
5	Ted	twilson	19.50	3

Degree - # of fields, ex. **degree = 5**

Cardinality - # of tuples/records, ex. cardinality = 3

Field Data types – numeric, character/text, date, logical

General Relational Model Terminology

Database → Relations → Tuples → Attributes

Employee Table:

emp#	Name	login	HourlyRate	NumDependents
3	Bob	bsmith	12.50	1
9	Carol	cjones	24.00	2
5	Ted	twilson	19.50	3

Degree - # of fields, ex. degree = 5

Cardinality - # of tuples/records, ex. **cardinality = 3**

Field Data types – numeric, character/text, date, logical

Functional Dependency

If given a value for attribute A, we know the value for attribute B, then A is said to identify B

$A \rightarrow B$

B is functionally dependent on A

Ex. $STU_NUM \rightarrow STU_LNAME$
 ↑ ↑
 determinant **dependent**

Functional Dependency - One-to-One Examples

StudentID \rightarrow FirstName

StudentID \rightarrow LastName

SSN \rightarrow Birthdate

StudentID \rightarrow Birthdate

StudentID, Course \rightarrow Grade

Functional Dependency - One-to-Many Examples

Father → Children

Bank → Customers

ShoppingCart → Items

Driver → Speeding Tickets

Patient → Appointments

Person → Computer Accounts

Person → Credit Cards

Functional Dependency - Many-to-Many Examples

Student \rightarrow Classes

Store \rightarrow Products

Person \rightarrow Vehicles

Dilemma - M:N relationships are not supported directly in the relational environment. They lead to data redundancies and/or data anomalies.

Instead use 1:M to resolve the M:N relationships

Unique Identifiers

Candidate Key

- Field or set of fields that uniquely identifies a record
- Cannot be NULL
- Must be minimal
- Underlined in Schema

Primary Key (pk)

- One field or one candidate key
- Ex. Employee_ID in EMPLOYEE, Skill_ID in SKILL

Composite Key

- Set of fields
- e.g., both Employee_ID & Skill_ID in CERTIFIED

Unique Identifiers - Candidate/Primary Key Examples

People: SSN, StudentID, Employee#, etc

Books: ISBN

Automobiles: Vehicle Identification Number (VIN)

House in U.S.: (number, street, city, state) or (number, street, zip).

Unique Identifiers - Candidate Keys - Relational Schema

TableName(attribute1, attribute2, ..., attributeN)

- the primary key is a composite key consisting of 'attribute1' and 'attribute2'.

Ex:

emp#	Name	login	HourlyRate	NumDependents
3	Bob	bsmith	12.50	1
9	Carol	cjones	24.00	2
5	Ted	twilson	19.50	3

Employee (emp#, Name, login, HourlyRate, NumDependents)

Unique Identifiers - Surrogate Key

A candidate key that automatically increments

Table name: SKILL

Skill_ID	Skill_Name	Skill_Description
100	Basic Database Management	Create and manage database user accounts.
110	Basic Web Design	Create and maintain HTML and CSS documents.
120	Advanced Spreadsheets	Use of advanced functions, user-defined functions, and macroing.
130	Basic Process Modeling	Create core business process models using standard libraries.
140	Basic Database Design	Create simple data models.
150	Master Database Programming	Create integrated trigger and procedure packages for a distributed environment.
160	Basic Spreadsheets	Create single tab worksheets with basic formulas
170	Basic C# Programming	Create single-tier data aware modules.
180	Advanced Database Management	Manage Database Server Clusters.
190	Advance Process Modeling	Evaluate and Redesign cross-functional internal and external business processes.
200	Advanced C# Programming	Create multi-tier applications using multi-threading
210	Basic Database Manipulation	Create simple data retrieval and manipulation statements in SQL.
220	Advanced Database Manipulation	Use of advanced data manipulation methods for multi-table inserts, set operations, and correlated subqueries.

Skill_ID could be a surrogate key where each new record's ID increments by 10

INSERT INTO SKILL (**SKILL_NAME**, **SKILL_DESCRIPTION**) VALUES ('Basic Database Management', 'Create and manage database user accounts.');

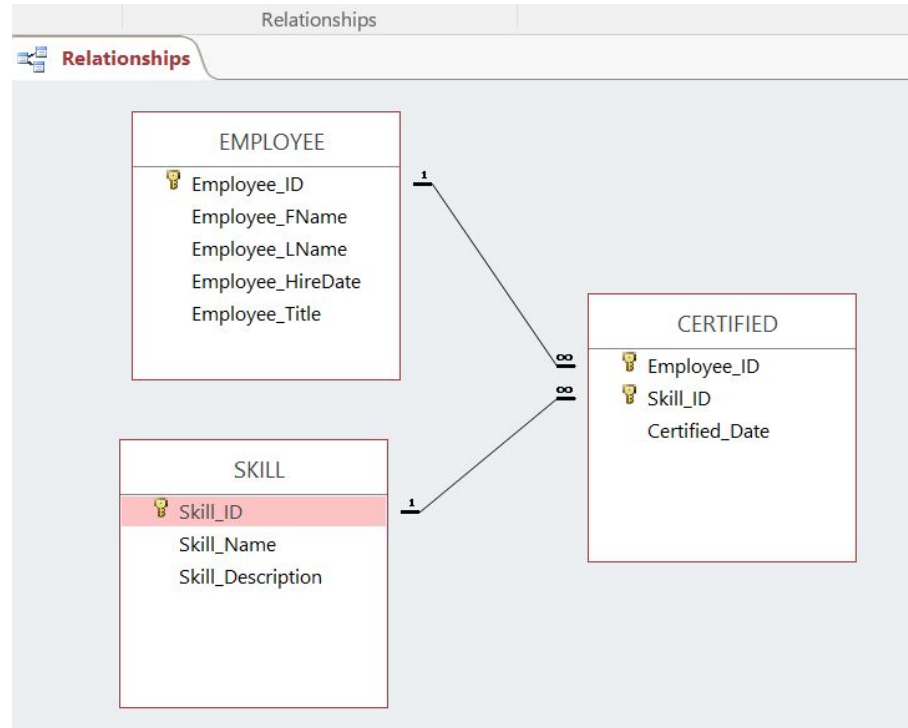
Unique Identifiers - Foreign Key

A record in one table **references a primary key** in a second table.

- Value must match a value in this second table or be null
- Relate tables using foreign keys
- Maintains data integrity by defining constraints
- Can be
 - optional (allows for a null entry)
 - mandatory (cannot be null)

Unique Identifiers - Foreign Key Example 1

What are the foreign keys?



Creating a Table with a Foreign Key

```
CREATE TABLE CERTIFIED (  
  EMPLOYEE_ID VARCHAR(6),  
  SKILL_ID    INTEGER,  
  CERTIFIED_DATE DATE,  
  CONSTRAINT PRIMARY KEY (EMPLOYEE_ID, SKILL_ID),
```

```
  CONSTRAINT fk1 FOREIGN KEY (EMPLOYEE_ID) REFERENCES  
  EMPLOYEE(EMPLOYEE_ID),
```

```
  CONSTRAINT fk2 FOREIGN KEY (SKILL_ID) REFERENCES SKILL (SKILL_ID)
```

```
);
```



Could also put outside of CREATE TABLE
***ALTER TABLE CERTIFIED ADD CONSTRAINT fk1 FOREIGN KEY
(EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID);***

Unique Identifiers - Foreign Key Example 2 - Lab 1

What are the foreign keys?

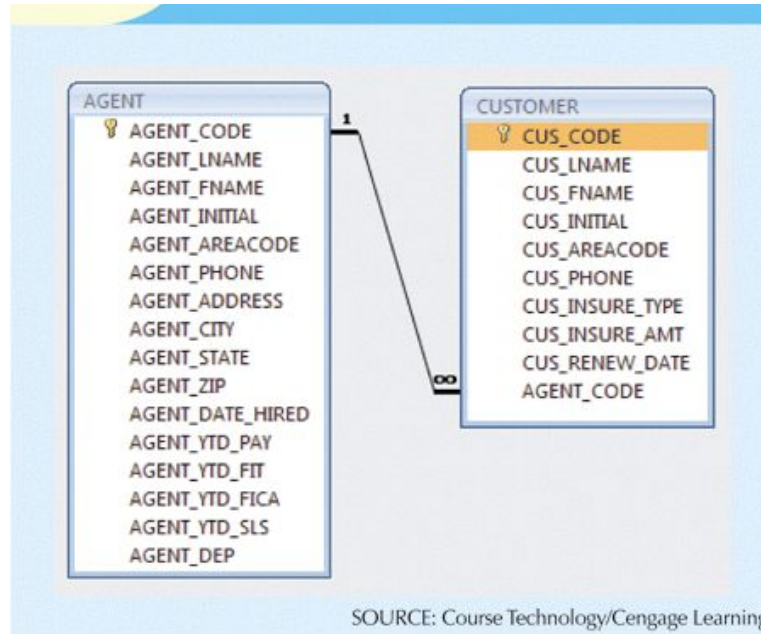
Students(StudentID, StudentName, Birthdate, Status, HighSchoolGPA)

Courses(CourseCode, Title, credits)

Enrollment(StudentID, CourseCode, Grade)

Unique Identifiers - Foreign Key Example 3

What are the foreign keys?



Unique Identifiers - Optional Foreign Key

May be null

Ex:

Patron(patronID, Name, Address, ...)

Books(ISBN, title, author, patronID)

- Books.patronID is a foreign key of Patron.patronID
- patronID is optional
- If Books.patronID is null, no one has checked out the book

Unique Identifiers - Mandatory Foreign Key

Is not null

Ex:

Driver (DLNum, Name, ...)

Officer (OfficerID, Name, ...)

Tickets (TicketNum, OfficerID, DLNum, Location, Speed, Date,...)

- Tickets.OfficerID is a foreign key to Officer.OfficerID
- Tickets.OfficerID is mandatory
- Do not allow a ticket to be written unless it has an OfficerID associated with it.

Unique Identifiers - Foreign Key - Referential Integrity

When referential integrity is enforced, a foreign key reference is required to have a match in the table being referenced.

Ex: Library book checkout.

Person/Patrons < -- >> Library Books (one person can check out many books)

Patron (patronID, Name, Address, ...)

Books (ISBN, title, author, patronID, ...)

Unique Identifiers - Foreign Keys - Homonyms & Synonyms

Homonym – 2 attributes with same name but different meanings

- ID & ID for Employee and Skill
- C_NAME & C_NAME for Client and Customer

Synonym – 2 attributes with different names but same meanings

- ID & Student_ID
- C_NAME & Name
- Prof_ID & Emp_ID

Unique Identifiers - Foreign Keys - M:M relationships

Handling foreign keys in M:M relationships

Ex:

Student << -- >> Classes

Student (SID, Name, Address, ...)

Courses (CourseID, Section, Semester, Room, Instructor, Date, Time)

Enrolled (SID, CourseID, Section, Semester, grade) - bridge/composite entity