

DATABASE DESIGN USING NORMALIZATION

revised by 김태연

OBJECTIVES

- To design updatable databases to store data received from another source
- To use SQL to access table structure
- To understand the advantages and disadvantages of normalization
- To design read-only databases to store data from updatable databases
- To recognize and be able to correct common design problems:
 - The multivalued, multicolumn problem
 - The inconsistent values problem
 - The missing values problem
 - The general-purpose remarks column problem

ASSESSING TABLE STRUCTURE

- General guidelines for assessing a table's structure are summarized:

- Count rows and examine columns
- Examine data values and interview users to determine:
 - Multivalued dependencies
 - Functional dependencies
 - Candidate keys
 - Primary keys
 - Foreign keys
- Assess validity of assumed referential integrity constraints

Copyright ©2014 Pearson Education

- Again, you can base your conclusions on sample data, but that data might not have all of the possible data cases.
- Therefore, verify your assumptions and conclusions with the users.

COUNTING ROWS AND EXAMINING COLUMNS

- To count the number of rows in a table:

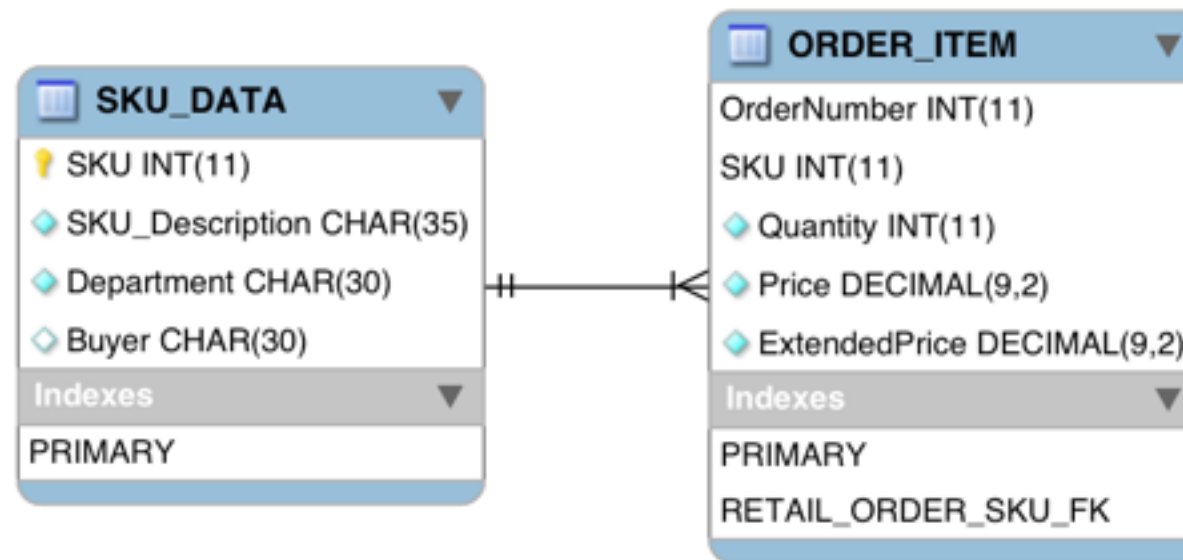
```
SELECT COUNT(*) AS NumRows  
FROM   SKU_DATA;
```

- To determine the number and type of columns in a table:

```
SELECT *  
FROM   SKU_DATA  
LIMIT 5;
```

CHECKING VALIDITY OF REFERENTIAL INTEGRITY CONSTRAINTS

- To find any foreign key values that violate the foreign key constraint:



```
SELECT    SKU
FROM      SKU_DATA
WHERE     SKU NOT IN
          (SELECT    SKU_DATA.SKU
           FROM      SKU_DATA, ORDER_ITEM
           WHERE     SKU_DATA.SKU = ORDER_ITEM.SKU) ;
```

NORMALIZATION – ADVANTAGES AND DISADVANTAGES

- On the positive side, normalization eliminates modification anomalies and reduces data duplication.
- On the negative side, normalization requires application programmers to write more complex SQL and read two or more tables.

- **Advantages**

- Eliminate modification anomalies

- Reduce duplicated data

- Eliminate data integrity problems

- Save file space

- **Disadvantages**

- More complicated SQL required for multitable subqueries and joins

- Extra work for DBMS can mean slower applications

COPYING DATA TO NEW TABLES

- You can create one table from another by adding a SELECT statement at the end of the CREATE TABLE statement:

ID	ItemNumber	EquipmentType	AcquisitionCost	RepairNumber	RepairDate	RepairCost
1	100	Drill Press	3500	2000	2013-05-05	375
2	200	Lathe	4750	2100	2013-05-07	255
3	100	Drill Press	3500	2200	2013-06-19	178
4	300	Mill	27300	2300	2013-06-19	1875
5	100	Drill Press	3500	2400	2013-07-05	0
6	100	Drill Press	3500	2500	2013-08-17	275

```
CREATE TABLE EQUIPMENT_ITEM
```

```
SELECT ItemNumber, EquipmentType, AcquisitionCost  
  
FROM EQUIPMENT_REPAIR;
```

ItemNumber	EquipmentType	AcquisitionCost
100	Drill Press	3500
200	Lathe	4750
300	Mill	27300

```
CREATE TABLE REPAIR
```

```
SELECT RepairNumber, ItemNumber, RepairDate, RepairCost  
  
FROM EQUIPMENT_REPAIR;
```

RepairNumber	ItemNumber	RepairDate	RepairCost
2000	100	2013-05-05	375
2100	200	2013-05-07	255
2200	100	2013-06-19	178
2300	300	2013-06-19	1875
2400	100	2013-07-05	0

READ-ONLY DATABASES

- Read-only databases are nonoperational databases using data extracted from operational databases.
- They are used for querying, reporting, and data mining applications.
- They are never updated (in the operational database sense—they may have new data imported from time to time).
- For example, Higher Education in Korea (<http://www.academyinfo.go.kr/>)

DENORMALIZATION

- For read-only databases, normalization is seldom an advantage.
- Application processing speed is more important.
- Denormalization is the joining of the data in normalized tables prior to storing the data.
- The data is then stored in nonnormalized tables.

```
CREATE TABLE EQUIPMENT_REPAIR
SELECT A.ItemNumber, A.EquipmentType, A.AcquisitionCost,
       B.RepairNumber, B.ItemNumber, B.RepairDate, B.RepairCost
FROM EQUIPMENT_ITEM A, REPAIR B
WHERE A.ItemNumber = B.ItemNumber;
```

COMMON DESIGN PROBLEMS

.....

- Although normalization and denormalization are the primary considerations when designing databases from existing data, there are four additional practical problems to consider.
- 1. Multivalued, Multicolumn Problem
 - VendorContact_1, VendorContact_2, VendorContact_3 ...
- 2. Inconsistent Values
 - Coded the same entries differently - GILDONG HONG, HONG GILDONG
 - When entries are misspelled - Coffee, Coffeee
- 3. Missing Values
 - A missing value, or null value, is a value that has never been provided.
- 4. General-Purpose Remarks Column
 - Columns with names such as Remarks, Comments, and Notes often contain important data that are stored in an inconsistent, verbal, and verbose manner.