Unit 3: The Relational Database Model

(Part 1)

Learning Objectives (1 of 2)

- In this chapter, you will learn:
  - That the relational database model offers a logical view of data
  - About the relational model's basic component: relations
  - That relations are logical constructs composed of rows (tuples) and columns (attributes)
  - That relations are implemented as tables in a relational DBMS

Recap: Logical vs Physical Data Format

Which of the following is Correct?

A

- Logical data format = how the human being views the data
- Physical data format = how the computer must work with the data

B

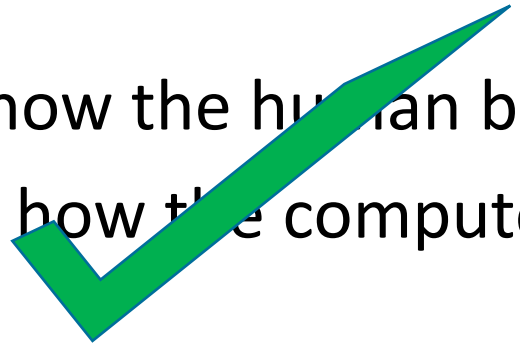
- Physical data format = how the human being views the data
- Logical data format = how the computer must work with the data

Recap: Logical vs Physical Data Format

Which of the following is Correct?

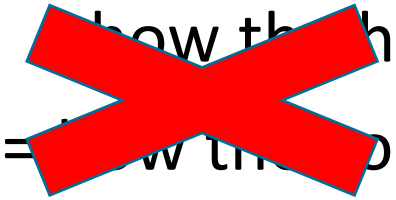
A

- Logical data format = how the human being views the data
- Physical data format = how the computer must work with the data



B

- Physical data format = how the human being views the data
- Logical data format = how the computer must work with the data



The Relational Model - Components

- The relational model has three well-defined components:
  - A logical data structure represented by relations
  - A set of integrity rules to enforce that the data is consistent and remains consistent over time
  - A set of operations that defines how data is manipulated

A Logical View of Data

- Relational database model enables logical representation of the data and its relationships
  - We need not view the physical representation
- Logical simplicity yields simple and effective database design methodologies
- Facilitated by the creation of data relationships based on a logical construct called a relation (table)

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.
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.

FIGURE 3.1 STUDENT TABLE ATTRIBUTE VALUES

Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1985	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92	No	ACCT	2315	230

STU_NUM = Student number
STU_LNAME = Student last name
STU_FNAME = Student first name
STU_INIT = Student middle initial
STU_DOB = Student date of birth
STU_HRS = Credit hours earned
STU_CLASS = Student classification
STU_GPA = Grade point average
STU_TRANSFER = Student transferred from another institution
DEPT_CODE = Department code
STU_PHONE = 4-digit campus phone extension
PROF_NUM = Number of the professor who is the student's advisor

Keys

- Consist of one or more attributes that determine other attributes
- Used to:
  - Ensure that each row in a table is uniquely identifiable
  - Establish relationships among tables and to ensure the integrity of the data
- **Primary key (PK):** Attribute or combination of attributes that uniquely identifies any given row

Determination

- State in which knowing the value of one attribute makes it possible to determine the value of another
- Basis for establishing the role of a key
- Based on the relationships among the attributes

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME}$$

Dependencies

- **Functional dependence**: Value of one or more attributes determines the value of one or more other attributes
  - **Determinant**: Attribute whose value determines another
  - **Dependent**: Attribute whose value is determined by the other attribute
- **Full functional dependence**: Entire collection of attributes in the determinant is necessary for the relationship

Exercise: Determinant vs Dependent

- Which is the Determinant ?
- Which is the Dependent ?

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME}$$

Exercise: Determinant vs Dependent

- Which is the Determinant ? – STU_NUM
- Which is the Dependent ? - STU_LNAME

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME}$$

Exercise: Functional Dependence

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME}$$

Complete these sentences

- _____ functional determines _____
- _____ is functionally dependent on _____

Exercise: Functional Dependence

$$\text{STU\_NUM} \rightarrow \text{STU\_LNAME}$$

Complete these sentences

- STU_NUM functional determines STU_LNAME
- STU_LNAME is functionally dependent on STU_NUM

Full Functional Dependence

- Full functional dependence
  - Functional dependencies in which the entire collection of attributes in the determinant is necessary for the relationship.

Exercise: Full Functional Dependence

- Which of these relationships exhibits full functional dependence

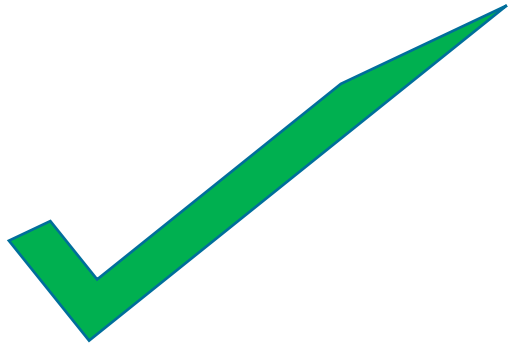
$$\text{STU\_NUM} \rightarrow \text{STU\_GPA}$$

$$(\text{STU\_NUM}, \text{STU\_LNAME}) \rightarrow \text{STU\_GPA}$$

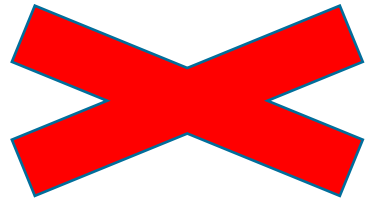
Exercise: Full Functional Dependence

- Which of these relationships exhibits full functional dependence

$$\text{STU\_NUM} \rightarrow \text{STU\_GPA}$$



$$(\text{STU\_NUM}, \text{STU\_LNAME}) \rightarrow \text{STU\_GPA}$$



Types of Keys (1 of 2)

- A key is an attribute or group of attributes that can determine the values of other attributes.
  - determinants in functional dependencies
- **Composite key**: Key that is composed of more than one attribute
- **Key attribute**: Attribute that is a part of a key
- **Entity integrity**: Condition in which each row in the table has its own unique identity
  - All of the values in the primary key must be unique
  - No key attribute in the primary key can contain a null

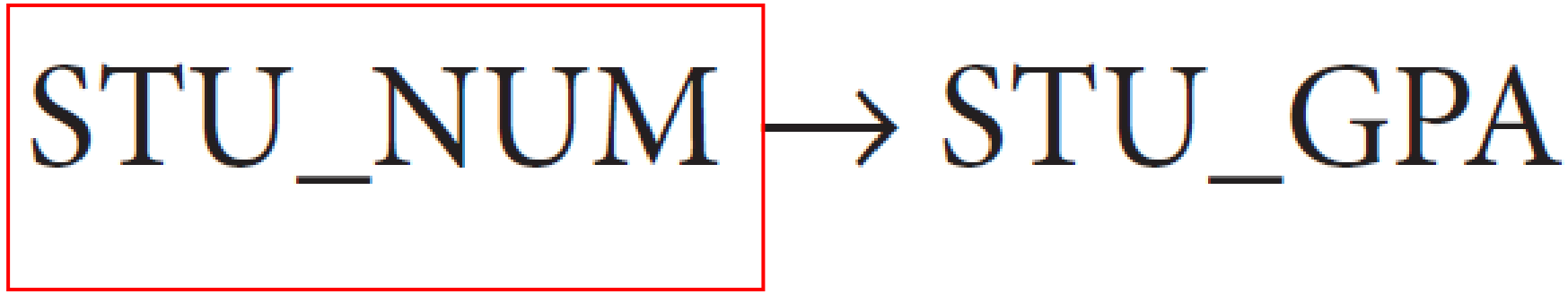
Exercise: Identify the Key(s)

- Identify the key in this functional dependency

$$\text{STU\_NUM} \rightarrow \text{STU\_GPA}$$

Exercise: Identify the Key(s)

- Identify the key in this functional dependency



Exercise: Identify the Key(s)

- Identify the key(s) in this functional dependency

$$(\text{STU\_LNAME}, \text{STU\_FNAME}, \text{STU\_INIT}, \text{STU\_PHONE}) \rightarrow \text{STU\_HRS}$$

Exercise: Identify the Key(s)

- Identify the key(s) in this functional dependency

$$(\text{STU\_LNAME}, \text{STU\_FNAME}, \text{STU\_INIT}, \text{STU\_PHONE}) \rightarrow \text{STU\_HRS}$$

Exercise: Identify the Composite Key

- Which functional dependency has a composite key?

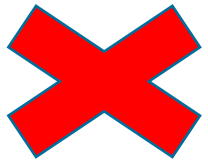
$$\text{STU\_NUM} \rightarrow \text{STU\_GPA}$$

$$(\text{STU\_LNAME}, \text{STU\_FNAME}, \text{STU\_INIT}, \text{STU\_PHONE}) \rightarrow \text{STU\_HRS}$$

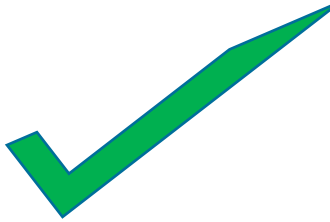
Exercise: Which is a Composite Key

- Which functional dependency has a composite key?

$$STU\_NUM \rightarrow STU\_GPA$$



$$(STU\_LNAME, STU\_FNAME, STU\_INIT, STU\_PHONE) \rightarrow STU\_HRS$$



Types of Keys (2 of 2)

- **Null**: Absence of any data value that could represent:
  - An unknown attribute value
  - A known, but missing, attribute value
  - A inapplicable condition
- **Referential integrity**: Every reference to an entity instance by another entity instance is valid
- **Secondary key:** Key used strictly for data retrieval purposes

Figure 3.2 - An Example of a Simple Relational Database

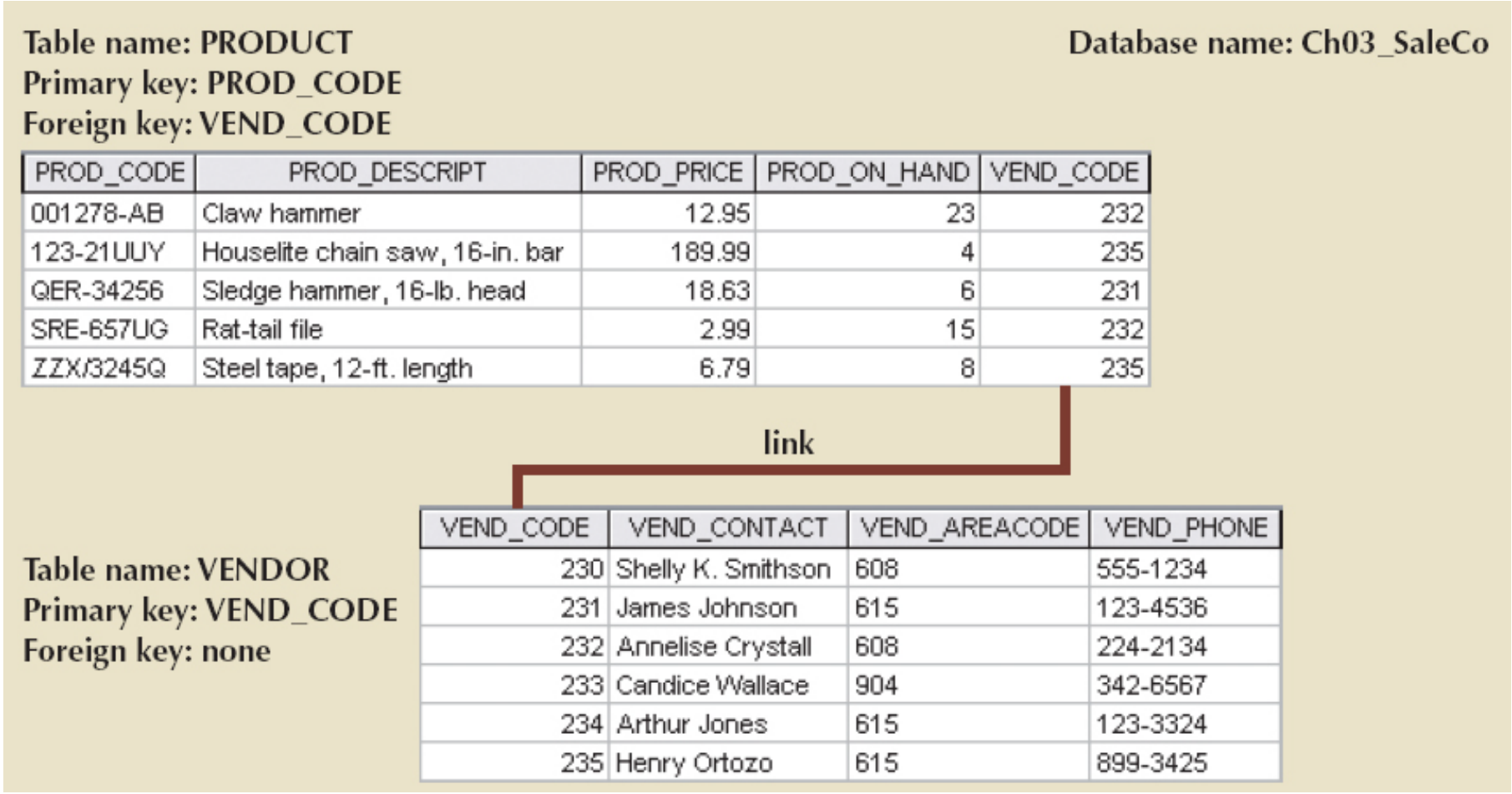


Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Table 3.3 - Relational Database Keys

KEY TYPE	DEFINITION
Super key	An attribute or combination of attributes that uniquely identifies each row in a table
Candidate key	A minimal (irreducible) super key; a super key that does not contain a subset of attributes that is itself a super key
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row; cannot contain null entries
Foreign key	An attribute or combination of attributes in one table whose values must either match the primary key in another table or be null
Secondary key	An attribute or combination of attributes used strictly for data retrieval purposes

Exercise: Identify the Super Key(s)

- An attribute or combination of attributes that uniquely identifies each row in a table

Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1985	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92	No	ACCT	2315	230

Exercise: Identify the Super Key(s)

- An attribute or combination of attributes that uniquely identifies each row in a table

Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1985	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1991	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1979	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1986	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1968	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1989	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1983	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1996	15	Fr	2.92	No	ACCT	2315	230

1. **STU_NUM**
2. **(STU_NUM, STU_LNAME),**
3. **(STU_NUM, STU_LNAME, STU_INIT),**
4. **(STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE).**

Exercise: Keys vs Super Keys

- Recall: A key consist of one or more attributes that determine other attributes

All Keys Are Super Keys

True or False

Exercise: Keys vs Super Keys

- Recall: A key consist of one or more attributes that determine other attributes

All Keys Are Super Keys

False

Why?

Exercise: Keys vs Super Keys

- Recall: A key consist of one or more attributes that determine other attributes

All Keys Are Super Keys

False

Not all keys uniquely identifies a row in a table

Exercise: Candidate Keys

- Which of the following are Candidate Keys?
  - STU_NUM
  - (STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE)
  - (STU_NUM, STU_LNAME)

Exercise: Candidate Keys

- Which of the following are Candidate Keys?
  - STU_NUM
  - (STU_LNAME, STU_FNAME, STU_INIT, STU_PHONE)
  - (STU_NUM, STU_LNAME)

(STU_NUM, STU_LNAME) is a superkey, but it is not a candidate key because
STU_LNAME could be removed and the key would still be a superkey.

Exercise: Candidate Keys

- Can a table have more than one candidate key?

Exercise: Candidate Keys

- Can a table have more than one candidate key?

YES!

Integrity Rules

TABLE 3.4

INTEGRITY RULES

ENTITY INTEGRITY	**DESCRIPTION**
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null; in short, all invoices are uniquely identified by their invoice number.
REFERENTIAL INTEGRITY	**DESCRIPTION**
Requirement	A foreign key may have either a null entry, as long as it is not a part of its table's primary key, or an entry that matches the primary key value in a table to which it is related; (every non-null foreign key value *must* reference an *existing primary* key value).
Purpose	It is possible for an attribute not to have a corresponding value, but it will be impossible to have an invalid entry; the enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory matching foreign key values in another table.
Example	A customer might not yet have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).

Figure 3.3 - An Illustration of Integrity Rules

Table name: CUSTOMER
Database name: Ch03_InsureCo
Primary key: CUS_CODE
Foreign key: AGENT_CODE

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	05-Apr-2016	502
10011	Dunne	Leona	K	16-Jun-2016	501
10012	Smith	Kathy	W	29-Jan-2017	502
10013	Olowski	Paul	F	14-Oct-2016	
10014	Orlando	Myron		28-Dec-2016	501
10015	O'Brian	Amy	B	22-Sep-2016	503
10016	Brown	James	G	25-Mar-2017	502
10017	Williams	George		17-Jul-2016	503
10018	Farriss	Anne	G	03-Dec-2016	501
10019	Smith	Olette	K	14-Mar-2017	503

Table name: AGENT (only five selected fields are shown)
Primary key: AGENT_CODE
Foreign key: none

AGENT_CODE	AGENT_AREACODE	AGENT_PHONE	AGENT_LNAME	AGENT_YTD_SLS
501	713	228-1249	Alby	132735.75
502	615	882-1244	Hahn	138967.35
503	615	123-5589	Okon	127093.45

Nulls

- The absence of an attribute value.
- Note that a null is not a blank.
- A null is no value at all.
- It does not mean a zero or a space.
- A null is created when you press the Enter key or the Tab key to move to the next entry without making an entry of any kind.
- Pressing the Spacebar creates a blank (or a space).

Ways to Handle Nulls

- **Flags**: Special codes used to indicate the absence of some value
- NOT NULL constraint - Placed on a column to ensure that every row in the table has a value for that column
- UNIQUE constraint - Restriction placed on a column to ensure that no duplicate values exist for that column