Chapter 3

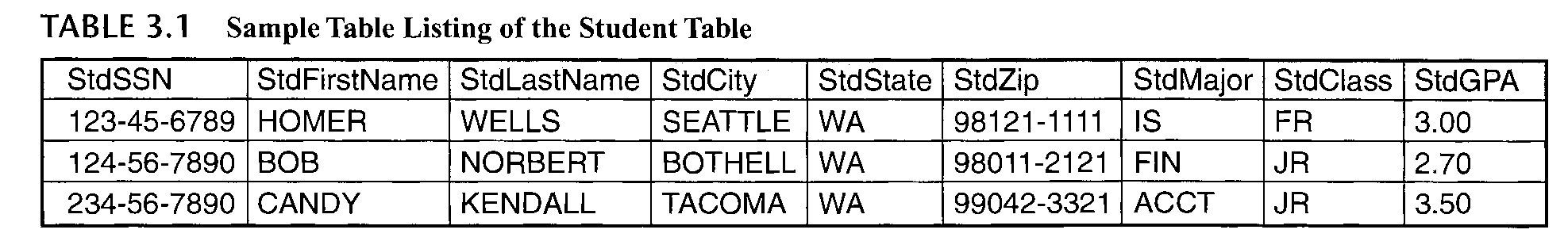
The Relational Data Model 45

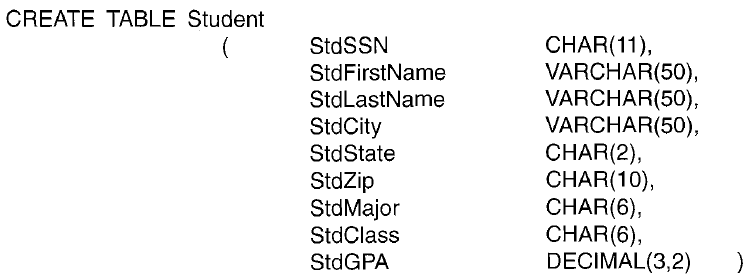
Learning Objectives 45

Overview 45

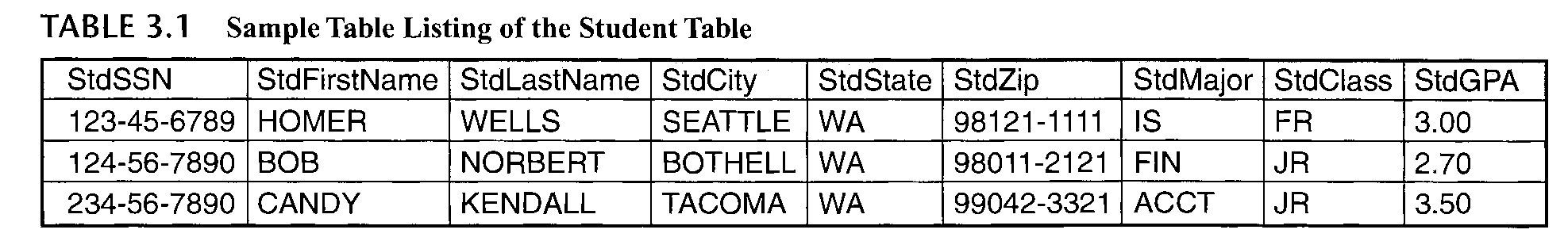
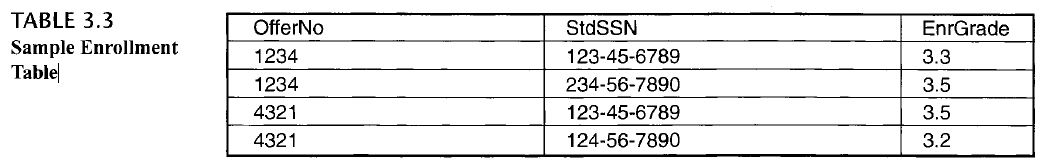
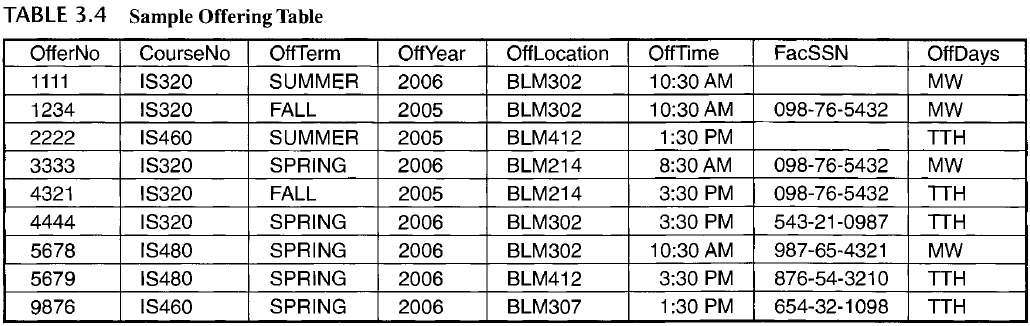
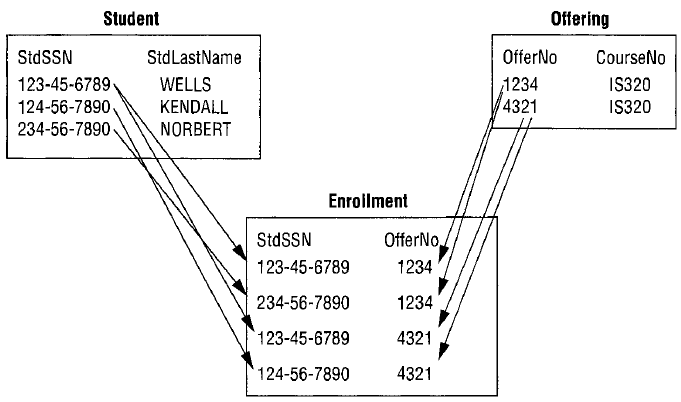
3.1 Basic Elements 46

3.1.1 Tables 46 – a two dimensional arrangement of data. A table consist of heading defining the table name and column names and a body containing rows of data

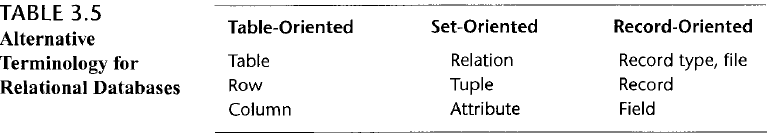


* Heading - part consists of the table name and the column names. This shows the column name
* Body – shows the rows of the table
* Data Type – defines a set of values and permissible operations on the values. Each column of a table is associated with a data type
  + CHAR(L) - For fixed-length text entries such as state abbreviations and Social Security numbers. Each column value using CHAR contains the maximum number of characters (/\_) even if the actual length is shorter. Most DBMSs have an upper limit on the length **(L)** such as 255.
  + VARCHAR(L) - For variable-length text such as names and street addresses. Column values using VARCHAR contain only the actual number of characters, not the maximum length for CHAR columns. Most DBMSs have an upper limit on the length such as 255.
  + FLOAT(P) - For columns containing numeric data with a floating precision such as interest rate calculations and scientific calculations. The precision parameter P indicates the number of significant digits. Most DBMSs have an upper limit on P such as 38. Some DBMSs have two data types, REAL and DOUBLE PRECISION, for low- and high-precision floating point numbers instead of the variable precision with the FLOAT data type.
  + DATE/TIME - For columns containing dates and times such as an order date. These data types are not standard across DBMSs. Some systems support three data types (DATE, TIME, and TIMESTAMP) while other systems support a combined data type (DATE) storing both the date and time.
  + DECIMAL(Wr R) - For columns containing numeric data with a fixed precision such as monetary amounts. The W value indicates the total number of digits and the R value indicates the number of digits to the right of the decimal point. This data type is also called NUMERIC in some systems.
  + INTEGER - For columns containing whole numbers (i.e., numbers without a decimal point). Some DBMSs have the SMALLINT data type for very small whole numbers and the LONG data type for very large integers.
  + BOOLEAN - For columns containing data with two values such as true/false or yes/no.
* 

3.1.2 Connections among Tables 47

* Relationships – connection between rows in a tables. Relationships are shown by column values in one table that match column values in another table
* Example
  + 
  + 
  + 
  + 
  + Relationship is like combining tables
  + Note: When columns have identical names in two tables, it is customary to precede the column name with the table name and a period as Student.StdSSN and Enrollment.StdSSN

3.1.3 Alternative Terminology 49



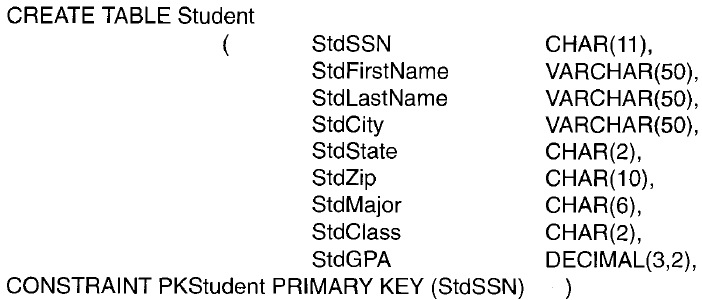
* table-oriented terminology appeals to end users
* set-oriented terminology appeals to academic researchers
* record-oriented terminology appeals to information systems professionals

3.2 Integrity Rules 49

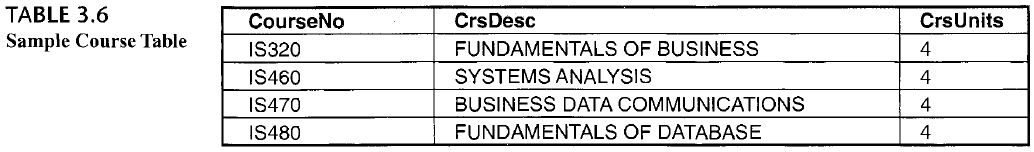
3.2.1 Definition of the Integrity Rules 49

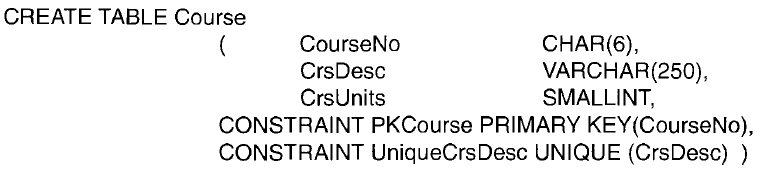
* Entity integrity means that each table must have a column or combination of columns with unique values
  + Unique means that no two rows of a table have the same value
    - For Example
      * StdSSN in Student is unique
      * StdSSN and OfferNo is unique in Enrollment
    - Business entities be easily traceable
  + also known as uniqueness integrity
* Referential integrity means that the column values in one table must match column values in a related table
  + Ensures that a database contains valid connections
  + For example, the value of StdSSN in each row of the Enrollment table must match the value of StdSSN in some row of the Student table
* Superkey: a column or combination of columns containing unique values for each row. The combination of every column in a table is always a superkey because rows in a table must be unique
* Candidate key: a minimal superkey. A superkey is minimal if removing any column makes it no longer unique.
* Null value: a special value that represents the absence of an actual value. A null value can mean that the actual value is unknown or does not apply to the given row.
* Primary key: a specially designated candidate key. The primary key for a table cannot contain null values.
* Foreign key: a column or combination of columns in which the values must match those of a candidate key. A foreign key must have the same data type as its associated candidate key. In the CREATE TABLE statement of SQL, a foreign key must be associated with a primary key rather than merely a candidate key.
* Integrity Rule
  + Entity integrity rule: No two rows of a table can contain the same value for the primary key. In addition, no row can contain a null value for any column of a primary key.
  + Referential integrity rule: Only two kinds of values can be stored in a foreign key: a value matching a candidate key value in some row of the table containing the associated candidate key or a null value.

3.2.2 Application of the Integrity Rules 50

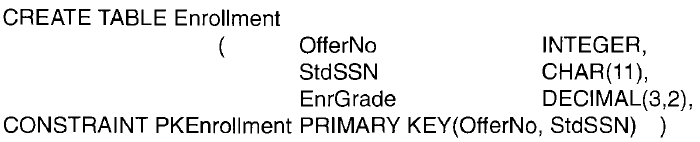


* Primary key – StdSSN
* CONSTRAINT keyword facilitates identification of the constraint if a violation occurs when a row is inserted or updated





* Entity Integrity Variations
  + Candidate keys that are not primary keys are declared with the UNIQUE keyword
  + Course table (see Table 3.6) contains two candidate keys: CourseNo (primary key) and CrsDesc (course description)
  + CourseNo column is the primary key because it is more stable than the CrsDesc column



* Enrollment table, the combination of StdSSN and OfferNo is the only candidate key
* composite or a combined primary key - A primary key consisting of more than one column

3.2.3 Graphical Representation of Referential Integrity 53

3.3 Delete and Update Actions for Referenced Rows 54

3.4 Operators of Relational Algebra 56

3.4.1 Restrict (Select) and Project Operators 56

3.4.2 Extended Cross Product Operator 57

3.4.3 Join Operator 59

3.4.4 Outer Join Operator 61

3.4.5 Union, Intersection, and Difference Operators 63

3.4.6 Summarize Operator 65

3.4.7 Divide Operator 66

3.4.8 Summary of Operators 68

Closing Thoughts 68

Review Concepts 69

Questions 69

Problems 70

References for Further Study 73