



Chapter 1 SQL and Data



What is Oracle?

- A relational database management system (DBMS)
 - software that allows the creation, retrieval and manipulation of data
- We will:
 - create new tables
 - define relationships between tables
 - add records
 - delete records
 - change field values so they're up-to-date
 - retrieve data to answer questions
 - control user access to data

2



What is SQL?

- Structured Query Language
- An industry-standard language used to access & manipulate data stored in a relational database
- E. F. Codd, 1970's IBM

3

3 Major Divisions of SQL

- Data Definition Language DDL (chapters 11, 12)
 - commands to create or alter database objects
 - tables, views, indexes, stored procedures
 - CREATE, ALTER, DROP
- Data Manipulation Language DML (chapters 2-10)
 - commands to retrieve or modify the contents of a table
 - SELECT, INSERT, UPDATE, DELETE
- Data Control Language DCL (chapter 14)
 - commands to control user privileges (access to database objects)
 - GRANT, REVOKE

4

Primary and Foreign Keys

- How do we connect (join) related data that has been stored in separate tables?
- Primary Key
 - 3 properties of:
 -
 -
 -
 - how indicated on schema diagram?
- Foreign key
 - how indicated on schema diagram?

5

Additional Primary Key Topics

- Candidate key (nib)
- Composite (concatenated) primary key
- Natural primary key
- Synthetic primary key

6



Columns and Datatypes

- When a table is created, each column is declared to contain data of a particular type
- Example datatypes:
 - varchar2(10)
 - char(10)
 - number(6,2)
 - date
- Domain
 - the set of permissible values from which a column draws its actual values

7



Schema

- A collection of **objects** owned by a database user and having the same name as the user
- Objects such as tables, views, indexes, sequences, synonyms, stored procedures, functions, triggers, packages
- Schema Diagram
 - a pictorial representation of a database's tables and its inter-table relationships
 - Student Database Schema (Appendix D)
 - Issue25 Database Schema

8



Data Redundancy

- What is it?
- Three main problems it causes:
 -
 -
 -
- Issue25 handout

9

One-to-Many Relationships (1:M)

- The most common type of inter-table relationship
- Each row in Table A can have **many** (zero or more) matching rows in Table B but each row in Table B has **only one** matching row in Table A.
 - each ZipCode can have many Students but each Student has only one ZipCode
 - each Course can be associated with many Sections but each Section covers only one Course
- Implementation
 - include the primary key of the one table as a foreign key in the many table
- How depicted in a schema diagram?
 - join line: crows feet and single line

10

Many:Many Relationships (M:M)

- Each row in Table A can have **many** matching rows in Table B & each row in Table B can have **many** matching rows in Table A.
 - each Pitcher can pitch in many Games and each Game can use many Pitchers
 - each Author can write many Books and each Book can be written by many Authors
- Figure 1.16, pg 20
- Implementation
 - create a third table (intersection table, junction table) and include each table's primary key as foreign keys
 - doing so breaks the Many:Many relationship into two separate 1:Many relationships
- How depicted in a schema diagram?
 - junction table and 2 join lines

11

One-to-One Relationships (1:1)

- Each row in one table has *at most* **one** matching row in the other table
- Tables whose primary keys are the same and also serve as foreign keys to the other table
- Columns in one table are additional attributes for the other table
- Figure 1.14, pg 18
- Commonly used to
 - avoid **nulls**
 - control field access

12



Join Lines

- Each join line runs between two tables and can be read in two directions
- Cardinality
 - indicates whether a table can have 0, 1, or Many matching rows in the related table
 - single line or crow's feet
- Optional/Mandatory Relationships
 - circle indicates optional relationship
 - vertical bar indicates mandatory relationship
- Figure 1.17, pg 21
 - each PUBLISHER **may** publish 0, 1, or many BOOKs
 - there can be publishers with no matching books
 - each BOOK **must** be published by 1 PUBLISHER
 - each book's PUBLISHER_ID must have a value (NOT NULL)

13



Database Development

- Requirements Analysis
 - identify user needs, data elements
- Conceptual Data Model
 - group data elements into entities; unique identifiers
- Logical Data Model
 - Normalization process: ERD: primary keys identified
 - designed to accommodate growth and change
- Physical Data Model
 - column names, datatypes, relationships, schema diagram
 - is platform specific
- Implementation
 - use SQL statements to create the tables and other objects

14



Referential Integrity

- Each foreign key value must match a primary key value in the related table
 - if a row in one table references a row in another, the referenced row must exist
- Rules to ensure that data stored in related tables stays "in synch" with each other
 - prevents *orphan* records in a related table
- How can we prevent null foreign keys?

15



Referential Integrity

- When **INSERT** a new record to a related table, a matching record must already exist in the primary table
 - e.g.: when add a new Section, the Course_No you enter must match an existing Course_No in the Course table
- Can't **UPDATE** an existing record's primary key value in the primary table when there are matching records the related table
 - e.g.: change a Student's StudentID
- Can't **DELETE** an existing record from a primary table when matching records exist in a related table
 - e.g.: delete a Student

16



Recursive Relationship

- A.k.a. Self-Referencing relationship
- A table containing a foreign key that relates to its own primary key
- How depicted in a schema diagram?

- Which table has a recursive relationship in the
 - Student schema?
 - Issue25 schema?

17
