

Structured Query Language

INTRODUCTION AND OVERVIEW

Background

SQL is a language for interfacing with relational databases and is a common interface to many systems and has become the industry standard.

SQL differs from third generation programming languages in that it allows users or programmers to tell the RDBMS “what” they want done without regard for “how” it is done.

This concept makes it suitable for use by non-programmers whose only concern is the updating and retrieval of information.

Standards

- Time spent learning the SQL interface on one supplier's RDBMS will not be lost if the company then adopts a rival RDBMS which itself supports SQL.
- The bulk of SQL is standard across all RDBMS platforms.
- Most vendors introduce non-standard extensions to the language to enhance the capabilities of their product.
- Non-standard extensions obviously lead to platform specific versions, but it does allow the American National Standards Institute (ANSI) to include the best extensions in the next version of the standard language.

SQL

Main language for relational DBMSs.

Main characteristics:

- relatively easy to learn;
- non-procedural - you specify *what* information you require, rather than *how* to get it;
- consists of standard English words like SELECT, INSERT, and UPDATE;
- can be used by range of users.

Importance of SQL

First and, so far, only standard database language to gain widespread acceptance.

Huge investment from both vendors and users.

Federal Information Processing Standard (FIPS).

Used as the basis for other standards.

Objectives of SQL

Ideally database language should let user:

- create database and table structures.
- perform basic tasks like insert, update, delete;
- perform both simple and complex queries.

Must perform these tasks with minimal user effort.

Must be easy to learn.

Objectives of SQL

SQL is a transform-oriented language with 2 major components:

- a DDL for defining database structure;
- a DML for retrieving and updating data.

SQL can be used interactively or embedded in a high-level language (eg. C, C++).

Writing SQL Commands

SQL statement consists of *reserved words* and *user-defined words*.

- Reserved words: fixed part of SQL and must be spelt exactly as required and cannot be split across lines.
- User-defined words: made up by user and represent names of various database objects such as tables, columns, views.

Writing SQL Commands

Most components of an SQL statement are *case insensitive*, except for literal character data.

Use extended form of BNF notation:

- Upper-case letters represent reserved words.
- Lower-case letters represent user-defined words.
- | indicates a *choice* among alternatives.
- Curly braces indicate a *required element*.
- Square brackets indicate an *optional element*.
- ... indicates *optional repetition* (0 or more).

Literals

Literals are constants used in SQL statements.

All non-numeric literals must be enclosed in single quotes (eg. 'London').

All numeric literals must not be enclosed in quotes (eg. 650.00).

SELECT Statement

```
SELECT [DISTINCT | ALL]
{* | [columnExprn [AS newName]] [...]}
FROM TableName [alias] [, ...]
[WHERE condition]
[GROUP BY columnList] [HAVING condition]
[ORDER BY columnList]
```

SELECT Statement

FROM Specifies table(s) to be used.

WHERE Filters rows.

GROUP BY Forms groups of rows with same column value.

HAVING Filters groups subject to some condition.

SELECT Specifies which columns are to appear in output.

ORDER BY Specifies the order of the output.

Looking at the SQL Command Categories

The three basic ANSI categories of command:

- Data Definition Language (DDL)
- Data Control Language (DCL)
- Data Manipulation Language (DML)

DDL Commands

DDL Commands

- These commands are used for creating, altering and dropping (deleting) database tables.
- The use of DDL commands is usually restricted to a Database Administrator (DBA) whose has overall control of the database.
- Typical DDL commands are:
 - Create Table Create View Drop Index
 - Alter Table Drop View
 - Drop Table Create Index

DCL Commands

DCL Commands

- DCL commands are used by the DBA to grant authorisation for data access, for allocating space and for auditing database use.
- Typical DCL commands are:
 - GRANT used to grant and limit privileges.
 - REVOKE used to remove privileges.

DCL Commands

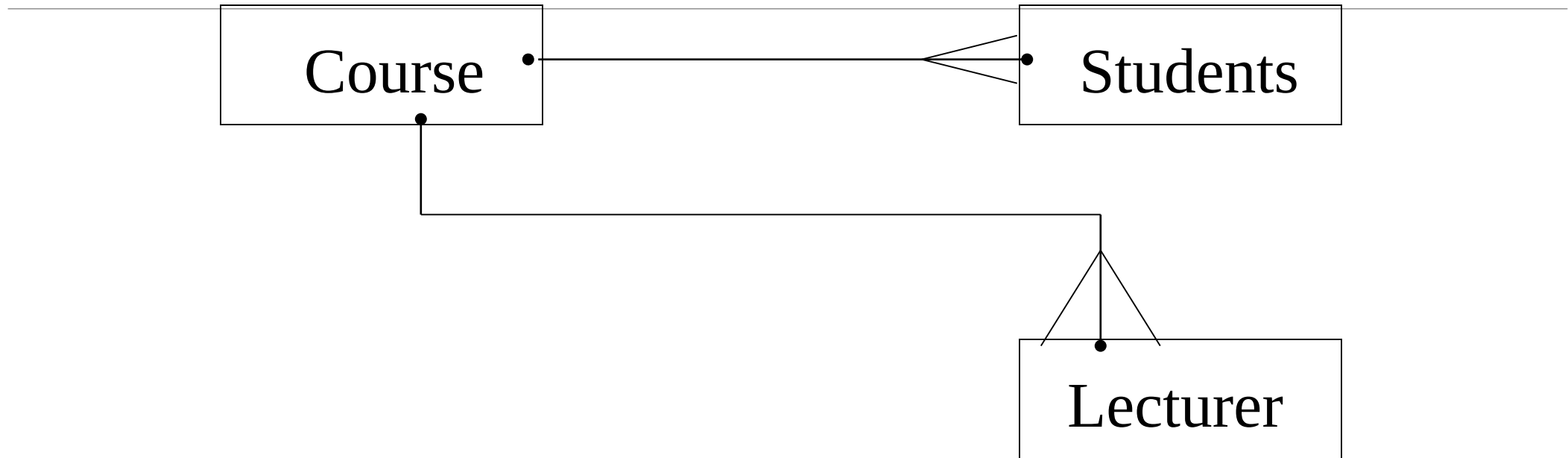
- COMMIT used to write changes to physical disk based database and therefore make them permanent.
- ROLLBACK if used before a COMMIT will reverse the set of changes that have been made, is often used in response to an error condition during database transaction processing.

DML Commands

DML Commands

- These commands are used for inserting, deleting and querying data values in the database.
- Typical DDL commands are:
 - SELECT
 - INSERT
 - DELETE
 - UPDATE

Trying out SQL with SQL College Case Study



In this Entity Relationship Model it can be said that a COURSE has many students, but a student can only be enrolled onto a single course.

A course will employ many LECTURERS, but a LECTURER is only associated with a single COURSE.

STUDENT TABLE

Student_No	Name	Address	Tel_No	Course_No
A001	Jones B	12 New Road	126152	HNC34
A016	Davies T	25 Powys Ave	417851	HNC60
A029	Lewis S	24 High Street	256985	HND12
B028	Harris K	94 Woodlands	201302	HNC34
C794	Dollard A	19 New Ave	500145	HNC34
C798	Jenkins L	12 Mount View	589410	HNC60
D927	Cook F	245 Neath Road	201630	HND12
E020	Compo T	1 Ferret Row	785012	HNC60
E029	Clegg N	6 Welly Street	541320	HND12
S999	Foggy C	5 Stocking Ave	323636	HND12

COURSE TABLE

Course_No	Course_Name	Course_Tutor
HNC34	HNC Business Studies	Rees I
HNC60	HNC Computer Studies	Biggs R
HND12	HND Electronics	Thatcher D

LECTURER TABLE

Lecturer_No	Name	Campus	Tel_Ext	Course_No
L263	Williams N	Town	2231	HNC34
L673	Griffiths D	Town	2303	HNC60
L246	Hope P	Mayals	2776	HNC60
L924	Stevens K	Sketty	2993	HND12
L930	Kendall H	Mayals	2776	HND12
L990	Biggles T	Town	2331	HND12

Create Table (Student)

Use the following commands to create the Student, Course and Lecturer Tables:

- **CREATE TABLE Student (Student_No TEXT(4) CONSTRAINT student_no_pk PRIMARY KEY,**
- **Name TEXT(25),**
- **Address TEXT(25),**
- **Tel_No TEXT(6),**
- **Course_No TEXT(5));**

Create Table (Course)

- **CREATE TABLE Course (Course_No text(5) CONSTRAINT course_no_pk PRIMARY KEY, Course_Name TEXT(25),**
- **Course_Tutor TEXT(20));**

Create Table (Lecturer)

- `CREATE TABLE Lecturer (Lecturer_No TEXT(4) CONSTRAINT lecturer_no_pk PRIMARY KEY,`
- `Name TEXT(25),`
- `Campus TEXT(20),`
- `Tel_Ext TEXT(4),`
- `Course_No TEXT(5));`

Insert Data Into Tables

Use the following commands to insert data into the Student, Course and Lecturer Tables:

- **INSERT INTO Student (Student_No, Name, Address, Tel_No, Course_No)**
- **VALUES ('A001', 'Jones B', '12 New Road', '469006', 'HNC34');**

Insert Data into Tables

Access changes the command to the one shown below.

- **INSERT INTO Student (Student_No, Name, Address, Tel_No, Course_No)**
- **SELECT 'A001', 'Jones B', '12 New Road', '469006', 'HNC34';**

Insert Data Into Tables

- `INSERT INTO Lecturer ( Lecturer_No, Name, Campus, Tel_Ext, Course_No )`
- `SELECT 'L263', 'Williams N', 'Town', '2331', 'HNC34';`
- `INSERT INTO Course ( Course_No, Course_Name, Course_Tutor )`
- `SELECT "HNC34", "HNC Business Studies", "Rees I";`

Populate The Database

Populate the database with the data given earlier before moving on to Data Manipulation Commands.

Select Command

Select all rows and all fields from the Student table:

- **SELECT ***
- **FROM STUDENT;**

Select all rows and a subset of fields from the Student table:

- **SELECT Student_No, Name, Course_No**
- **FROM STUDENT;**