SQL (MySQL)

School of Computer Science, UCD

Scoil na Ríomheolaíochta, UCD

Outline

- From Conceptual to Relational
- Introduction to SQL (MySQL)
- Records Manipulation in SQL (MySQL)

Take home message:

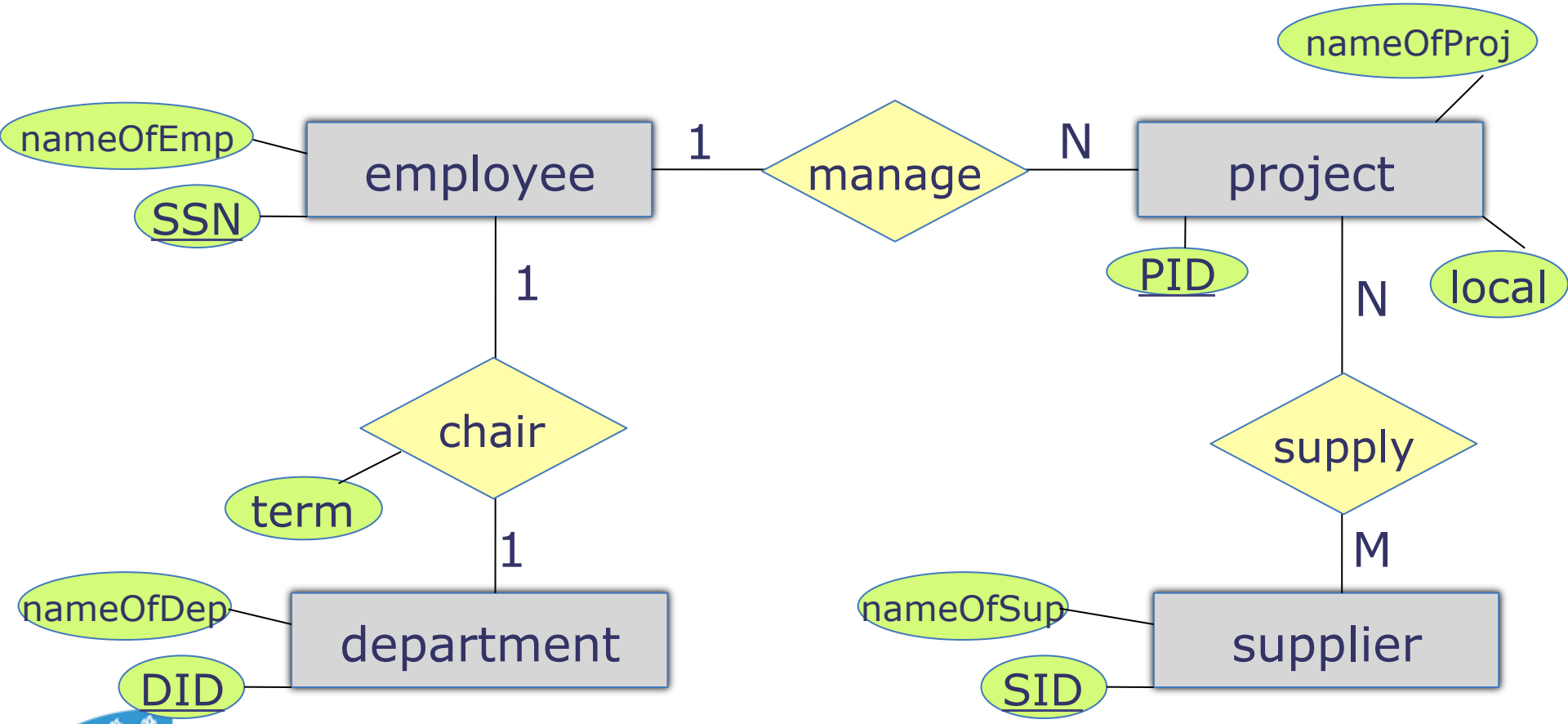
SQL is a declarative programming language to manipulate data

FROM CONCEPTUAL TO RELATIONAL

General Algorithm

1. Every entity becomes a relation with a key
2. Relationship 1:N sets a key to the N relation
3. Relationship N:M creates a new relation with the keys from both sides

Example


nameOfProj
nameOfEmp
employee
1
manage
N
project
SSN
PID
local
1
N
chair
supply
term
1
M
nameOfDep
department
nameOfSup
supplier
DID
SID

Example

Table: employee

Attr.	Type
SSN	integer
nameOfEmp	string

Table: supplier

Attr.	Type
SID	integer
nameOfSupp	string

Table: project

Attr.	Type
PID	integer
nameOfproject	string
local	string
SSN	integer

Table: supply

Attr.	Type
quantity	integer
product	string
PID	integer
SID	integer

INTRODUCTION TO MYSQL

MySQL

- Most popular open source database server
  - Performance
  - Reliability
  - Easy installation and maintenance
    - Provides robust set of SQL syntax elements
- Supports several well-known services
  - Facebook, Yahoo, Google, YouTube, etc
- It's the M of LAMP
  - Linux, Apache, MySQL, PHP
- Runs in almost any operating system

Products

- MySQL Community Server
  - Open and free
- MySQL Enterprise
  - Commercial license
  - Monitoring capabilities
- MySQL Cluster
  - Fault tolerance and performance
- MySQL Embedded Database
  - Embedded devices
- MySQL Workbench
  - Graphical tool for development and management

Install

- Installers for Linux, MacOS, OpenSolaris, Windows
  - Already available in most Linux distributions
- Deamon mysqld

Command Line Client

- Tool: mysql
- Available in any installation
- Connects with local and remote database servers
  - Default connection is local, using the same linux user currently connected
- Executes in interactive or batch modes
  - Scripts execution
  - Can be scheduled using crontab
  - In interactive mode, it is commad shell for SQL
- Several options
  - $ mysql --help

Command line options

- Connecting using default user (currenty connected) in localhost
  - $ mysql
- Connectiong as named user with password in localhost
  - $ mysql –u *username* –p
- Connecting in a given host with named user and password
  - $ mysql –h *192.168.1.100* –u *username* -p

What is SQL?

- Language for data retrieving and manipulation, in relational databases
  - Data definition
  - Data Manipulation
  - Data control and security
- Open standard – ANSI
  - Nonetheless, providers implement proprietary features

SQL Terminology

- Table
  - Set of rows or records
  - Similar to a “data file”
- Row
  - Similar to a record of an “data file”
- Column
  - Record field, registered information
  - Each column in a given row has a unique value
- Primary key
  - One or more columns whose contents are unique in a table
  - Identify a record of a table

SQL command types

- DDL
  - Data Definition Language
    - Create and modify table and object structure
    - Ex.: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE VIEW, etc
- DML
  - Data Manipulation Language
    - Manipulate table data
    - Ex.: INSERT INTO, UPDATE, DELETE, SELECT
- DCL
  - Data Control Language
    - Control user access
    - Ex.: GRANT, REVOKE

SQL: databases

- List the available databases
  > SHOW DATABASES;
- Create a database
  > CREATE DATABASE database1;
- Set a database as current
  > USE database1;
- Delete a database
  > DROP DATABASE database1;
  > DROP DATABASE IF EXISTS database1;

SQL: tables

- List tables
  > SHOW TABLES;
- Create tables
  - CREATE TABLE
    - Table name
    - Field list, including type, size and modifiers
    - Primary key
    - Database engine

SQL: tables

- Create table

```
> CREATE TABLE contacts (
    contact_id SMALLINT NOT NULL,
    name VARCHAR(45) NOT NULL,
    address VARCHAR(50),
    age SMALLINT UNSIGNED,
    PRIMARY KEY (contact_id)
    );
```

SQL: tables

- Display table details
  > DESCRIBE contacts;
- Alter name and type of a field
  > ALTER TABLE contacts CHANGE name full_name varchar(50);
- Add field
  > ALTER TABLE contacts ADD city varchar(30) not null;
- Remove field
  ALTER TABLE contacts DROP age;
- Delete table
  > DROP TABLE table1;
  > DROP TABLE IF EXISTS table1;
- Using tables from multiple databases
  > SELECT * FROM database1.table1;

Field definition

- Definition for each field
  - Name
  - Typo
  - Modifier or restriction

Field types

- TINYINT
  - 1 byte, -128 to 127 (signed), 0 to 255 (unsigned)
- SMALLINT
  - 2 bytes, -32768 to 32767 (signed), 0 to 65535 (unsigned)
- MEDIUMINT
  - 3 bytes
- INT
  - 4 bytes
- BIGINT
  - 8 bytes

Field types

- FLOAT
  - Single precision
- DOUBLE
  - Double precision
- DECIMAL
  - Decimal values
- BIT
  - Bit values
    - b’0101’

Field types

- CHAR
  - Fixed String up to 255 characters
- VARCHAR
  - Variable length String up to 255 characters
- DATE, TIME, YEAR
- DATETIME, TIMESTAMP
- ENUM, SET
  - Predefined set of values

Field modifiers

- NULL or NOT NULL
  - Allows (or not) null values
- DEFAULT
  - Default value, if not specified in INSERT
- AUTO_INCREMENT
  - Database generates incremental number automatically
  - Used to generate unique primary keys
- CHARACTER SET
  - Set of characters for an string

RECORDS MANIPULATION

Insert records

- Insert
- Insert one or multiple records
- Field list is optional
  - If not cited, the field natural order is used
- Exemple:

```
INSERT INTO contacts (contact_id, name, age)
VALUES (1, 'Contact 1', 25);

INSERT INTO contacts (contact_id, name, age)
VALUES
(2, 'Contact 2', 30),
(3, 'Contact 3', 35);
```

Delete records

- Delete
- “where” clause
- Exemple:

```
DELETE FROM contacts WHERE age < 10;
```

Update records

- Update
- Fields to be altered
- “where” clause
- Exemple:

```
UPDATE contacts SET age = 45
WHERE age = 99 OR name = ‘Anna’;
```

Display records

- Select
- Fields, where, joins, order etc
- Exemples:
  - Selecting fields

```
SELECT contact_id, name FROM
  contacts;
```

  - Filtering records

```
SELECT name, age FROM contacts
  WHERE age > 50;
```

Retrieve records

- Exemples:
  - Searching records in varchar/char fields using LIKE

```
SELECT name, age FROM contacts
  WHERE age < 50 AND name LIKE
  'Joseph%';
```

  - Sorting results

```
SELECT name, age FROM contacts
  ORDER BY age ASC;
```

    - ASC or DESC
  - Limiting the number of records in results

```
SELECT * FROM contacts LIMIT 10;
```

Arithmetic functions

- Basic operators

```
SELECT 8 + 3;
SELECT 54 * (23 + 2);
```

- Record count
  - COUNT

```
SELECT COUNT(*) FROM contacts;
SELECT COUNT(*) FROM contacts WHERE age >
  30;
```

Arithmetic functions

- Average
  - AVG

```
SELECT AVG(age) FROM contacts;
SELECT AVG(price) FROM books WHERE type_id =
  3;
```

- Sum
  - SUM

```
SELECT SUM(salary) FROM employee WHERE
  dept_id = 4;
```

Arithmetic functions

- Maximum
  - MAX

```
SELECT MAX(age) FROM contacts;
SELECT MAX(price) FROM books WHERE type_id =
  3;
```

- Minimum
  - MIN

```
SELECT MIN(age) FROM contacts WHERE name
  LIKE 'Anna%';
```