DATABASES & SQL

CS2141 - Software Development using C/C++

What is a Database?

* “A collection of data for one or more multiple uses”

* “A structured set of data held in a computer, esp. one that is accessible in various ways”

* System for storing & retrieving data

    * Hash tables, relational databases, “NoSQL”

Relational Databases

* First appeared around 1970
* Data stored in user-editable tables
* Tables are related to one another according to defined rules
* Very, very popular
    * Popular implementations by Oracle, Microsoft, IBM, MySQL, PostgreSQL and SQLite

Tables

* Collection of similar information (eg customer data, orders)
* Consists of records (rows) made up of fields (columns)
* Number of columns is fixed, rows are unlimited
* Tables with related data (eg customers and orders) are linked by columns with identical data (eg a customer id code)

Table Example

customers

custid	fname	lname	orders
jqp	John	Public	2
jjd	John	Doe	0
jas	John	Smith	1

orders

orderid	custid	order date	itemid
1	jqp	2 Oct 09	1
2	jas	14 Nov 09	2
3	jqp	24 Dec 09	2

items

itemid	desc	price
1	Staples	0.50
2	Envelopes	1.00

SQL

* Structured Query Language
* Standard language for interacting with databases
  * Retrieving/adding/updating/deleting records
* Also provides functionality for modifying database structure
* Core language is fixed, additional functions vary by vendor
* Case insensitive, but keywords are traditionally capitalized

SELECT

* Retrieves data, but does not change the database

*
```
SELECT column1[, column2...]
  FROM table1[,table2...]
  [WHERE condition]
  [ORDER BY column [DESC]];
```

* `WHERE` - Only return records where condition is true

* `ORDER BY` - Sort records by column

SELECT Examples

* Fetch all columns from all records in “customers”

```
SELECT * FROM customers;
```

* Get all columns for customer with id 2:

```
SELECT * FROM customers
WHERE custid LIKE ‘jqp’;
```

* Get customer names sorted by number of orders:

```
SELECT fname, lname, orders
FROM customers
ORDER BY orders;
```

Relationships & SELECT

* Two methods: JOIN and WHERE
  * JOIN is ‘correct’, WHERE easier for our purposes
*
```
SELECT fname, lname, orderdate, itemid
FROM customers, orders
WHERE customers.custid = orders.custid;
```

UPDATE

* Updates one or more existing records

```
UPDATE table
  SET column = new_value[, column = new_value]
  [WHERE condition];
```

* Be careful! Forgetting the `WHERE` will update all of the records

INSERT

* Add a new record to a table

*
```
INSERT INTO table (column1, [column2, ... ])
  VALUES (value1, [value2, ...]);
```

* Shortcut form (values must be in table order):

*
```
INSERT INTO table
  VALUES (value1, [value2, ...]);
```

* Defaults are used if a new value isn't specified

DELETE

* Removes records from a table

* ```
  DELETE FROM table
    [WHERE condition];
  ```

* Will delete all table values if `WHERE` is omitted
  ```