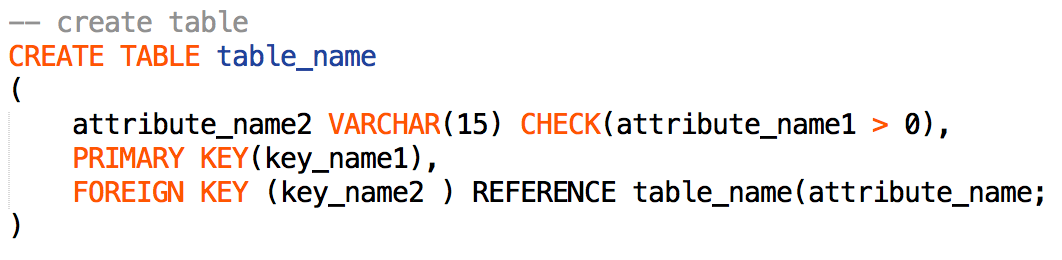
**X. Others**

1. **JVM**

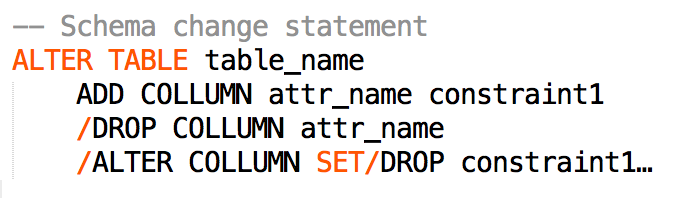
* **Concept**
* The Java virtual machine (JVM) is the platform upon which your programs run
* Java programs are compiled (using javac) into bytecode. This is interpreted by the JVM into the specific instructions for that architecture and operating system.
* **How is memory allocated?**
* The new keyword allocates memory on the Java heap. The heap is the main pool of memory, accessible to the whole of the application.
* If there is not enough memory available to allocate for that object, the JVM attempts to reclaim some memory from the heap with a garbage collection.
* If it still cannot obtain enough memory, an OutOfMemoryError is thrown, and the JVM exits.
* **What is garbage collection?**
* Garbage collection is the mechanism of reclaiming previously allocated memory, so that it can be reused by future memory allocations.
* **What is the difference between the stack and the heap?**
* Memory is split into two major parts, the stack and the heap.
* The stack is the place where any primitive values, references to objects, and methods are stored.
* Stack is used for static memory allocation and Heap for dynamic memory allocation
* The lifetime of variables on the stack is governed by the scope of the code
* When you call a method, declared variables are placed on top of the stack
* If a method calls itself too many times, the stack memory fills up, and results in a StackOverflowError
* Recursive methods usually use much more stack space

1. **SQL**

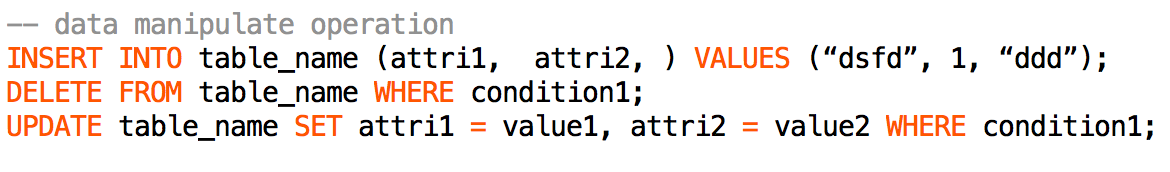
* **SQL statement syntax**
* **Create table**

****

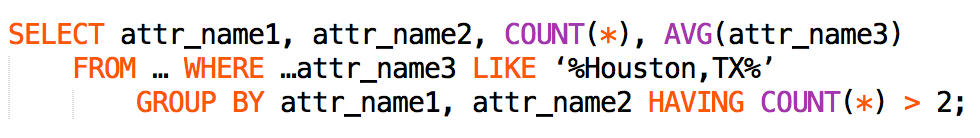
* **Schema change statement**

****

* **Data manipulate operation**

****

* **Join**
* LEFT JOIN keyword returns all rows from the left table with the matching rows in the right table
* NATURAL JOIN is a join assumes the join criteria to be where same-named columns in both table match
* **Aggregate Function**
* AVG, COUNT, SUM, MIN, MAX, etc.
* **View**
* A virtual table, and can be select from view as a table
* **Group By**



* **Important**: attribute which not named in the GROUP BY clause, cannot be used in the SELECT list or HAVING clause unless enclosed in an aggregate function.
* **Order**

**../../../../../Desktop/Screen%20Shot%202017-01-20%20at%2017.29.5**

* **Important1**: ORDER BY items must appear in the select list if SELECT DISTINCT is specified.
* **Important2:** ORDER BY items must appear in the select list unless enclosed in aggregate function if GROUP BY is specified.
* **Nested Query**
* WHERE attr\_name IN (SELECT attr\_name1…) OR IN
* WHERE attr\_name < ALL/ANY (SELECT …);
* WHERE EXISTS/NOT EXISTS (SELECT …);

(always use NOT EXIST rather than NOT IN, since when the subquery returns even one null, NOT IN will not match any rows)

* **Denormalized VS Normalized Database**
* Normalized databases are designed to minimize redundancy (store once but with drawback of expensive joins when many common queries)
* while denormalized databases are designed to optimized read time (usually large database)
* **ACID**
* ACID is a set of properties of database transactions.
* Atomicity: If one part of the transaction fails, then the entire transaction fails, and the database state is left unchanged
* Consistency: This property ensures that any transaction will bring the database from one valid state to another.
* Isolation: The effects of an incomplete transaction might not even be visible to another transaction.
* Durability: This property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.
* **Query Tricks**
* **Get duplicate**

GROUP BY xxx HAVING COUNT(\*) > 1

(only HAVING clause can use aggregate functions, not WHERE)

* **Compare Employee and Manager’s Salary**

SELECT xxx FROM Employee as e JOIN Employee as m ON …

* **Customer Who Never Order**

SELECT xxx FROM xxx WHERE c.id NOT EXIST (SELECT CustomerId from Orders)

* **Rising Temperature**

WHERE DATEDIFF(cur.DATE, pre.DATE)=1

* **K Highest Salary**

ORDER BY Salary DESC LIMIT K-1,1

* **Rank Score**

SELECT Score, (SELECT count(distinct Score) FROM Scores WHERE Score >= s.Score) Rank FROM Score s …

* **Department Highest Salary and Employee**

Just Select from Employee, Department, and self-created Department\_Highest\_Salary table, then joint them

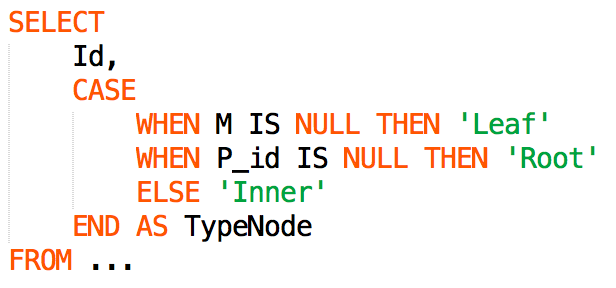
* **Department Top Three Salaries (cannot just use MAX(xxx))**

SELECT xxx FROM xxx WHERE

(select count(distinct(Salary)) from Employee WHERE

DepartmentId = E.DepartmentId and Salary > E.Salary) in (0, 1, 2)

* CASE in SELECT



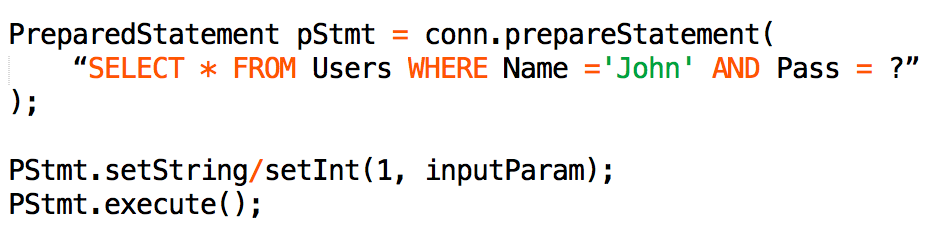
* Get Top K items use: LIMIT K at the end of query
* **NoSQL**
* do not necessarily follow the relational database model
* They often rely on denormalized data and are stored as key-value pairs, whole documents, or graphs in order to make retrieval as quickly as possible, particularly with huge amounts of data
* Popular NoSQL databases
* MongoDB – Data is stored as JSON objects, and may not share all of the same fields as other similar objects.
* Cassandra – Data is stored in column families, similar to tables, but rows do not need to share the same columns.
* **JDBC**
* Java Database Connectivity, the mechanism built into the standard Java libraries to connect to a database.
* Data Manipulation Language (DML) : SELECT, INSERT, UPDATE, and DELETE
* Data Definition Language (DDL) : CREATE and manipulate table structures.
* **Avoid SQL injection attack**
* **Risk**: hacker put password = " or ""=", and lead to:

SELECT \* FROM Users WHERE Name ="John"

AND Pass =" " or ""=" "

* **Solution**: use query parameters

The SQL engine checks each parameter to ensure that it is correct for its column and are treated literally, and not as part of the SQL to be executed.



* **How Java perform SQL queries**
* Statement stmt.execute (“use book\_library”);
* Stmt.executeUpdate(DML or DDL);
* stmt.executeQuery(“select…”);

1. **HTTP and REST**

* **HTTP**
* **Concept**: core way for requesting and receiving data over the Internet
* **Methods**: instructions to the web server
* GET—Serve the requested resource; no changes should be made.
* POST—Update the resource to the content of the request.
* PUT—Set the content of the resource to the content of the request.
* DELETE—Remove the requested resource.
* HEAD—Mimic a GET request, but only return the response code and headers.
* **REST**
* ***Representational State Transfer***
* use HTTP to provide remote API calls between systems
* Spring MVC carry the capability to extract variables from a request path, using **HttpMessageConverter**
* **SOAP**
* Simple Object Access Protocol
* **Target**
* SOAP provides a way to communicate over HTTP between applications running on different operating systems, with different technologies and programming languages.
* SOAP is an application communication protocol
* SOAP is a format for sending and receiving messages
* SOAP is platform independent
* SOAP is based on **XML** (eXtensible Markup Language)
* XML, as JSON was designed to store and transport data

1. **Scalability & Memory Limits**

