Lesson 2: Key takeaways

- SQL language: to interact with relational databases
    - SELECT for queries
    - DML for Data Manipulation Language (INSERT/UPDATE/DELETE)
    - DDL for Data Definition Language (CREATE TABLE, etc …)
    - Commit, Rollback (to savepoint) for Transaction Control
    - GRANT and REVOKE for Privileges Control

- Communication with the database:
    - A connection request is sent to the database listener, using a connect string
        - Database listener is a process running on the database machine, listening on a port (1521 by default for Oracle Databases, other port can be added) for incoming connections
        - Listener checks whether the request is correct (host, port and service name)
        - If the request is correct, a direct connection between the app and the database is started through a server process (process running on the database machine)

- Running a SQL statement against the database:
    - Three steps in the database: parse, execute and fetch (fetch only for SELECT statements returning rows to the app layer)
    - Parse: three sub-steps
        - Syntactic analysis: is the statement syntax correct ?
        - Semantic analysis: do the objects in the statement exist ? do you have the necessary privileges to access them ? **Database dictionary is used during this phase**.
        - Execution plan construction: find the best possible plan to solve the query. Multiple plans are evaluated during parse, each plan is costed, and the Optimizer chooses the cheapest. Optimizer highly relies on statistics collected on data structures (number of rows, distribution of the values in the columns, ...). When a plan is agreed to be optimal, it is used to execute the SQL sentence.
        - Parse result in cached in memory for further executions *of the same statement*.
    - Execute: execute the statement **using the calculated execution plan**. Collects execution statistics and compare them with the estimation made during parse, to adjust the plan for the next executions.
    - Fetch: fetches the rows into a result set, and return it to the application layer (in the SELECT case).

- Common architectures:
    - 3 layers architectures: end-user, application layer, database layer
        - Application layer usually composed **by several servers in cluster**, to ensure high availability (HA)
        - Database might be clustered as well (will be detailed in further lessons) to ensure HA.

- Evolution of Data Management and JSON support in RDBMS:
    - New tendencies eventually end integrated inside the RDBMS: Object Oriented models, XML, JSON .... until the next big thing.
    - JSON is very popular among developers. Databases storing data in JSON documents are popular, because it’s simple to interact with them (query on read).
    - JSON fully supported in RDBMS: store JSON documents into a table column. SQL has been enhanced with JSON specific constructs, to query JSON documents.

- RDBMS can be queried using REST API: URL are translated to SQL (simple cases) or mapped to SQL statements (more complex queries). This allows to have development teams specialized in SQL, that will provide only URL to non SQL specialized teams (**Abstract the application logic from SQL language**).