Amazon RDS (315)

https://aws.amazon.com/rds/

- Amazon handles setup, patching, backups, and capacity can be resized.
- Available in both MySQL and Oracle engines.
- Can translate the database to Amazon Aurora later if needed.
- Used by Netflix, General Electric, and Expedia, many others
- Used in web and mobile applications.

Amazon Aurora

https://aws.amazon.com/rds/aurora/?aurora-whats-new.sort-by=item.additionalFields.postDateTime&aurora-whats-new.sort-order=desc

- MySQL and PostgreSQL compatible.
- Also a fully managed service (automates hardware provisioning, setup, patching, backups)
- Capable of backups to Amazon S3
- Supports network isolation using VPC, can encrypt data in transit using SSL, data underlying can also be encrypted.
- No free version instance of Amazon Aurora.

Oracle (322)

https://www.sqlbot.co/blog/sql-server-vs-oracle
https://hevodata.com/learn/oracle-vs-sql-server/#pc

- SQL Server recently has Linux support as of 2017 (Oracle already had Linux support before).
- Oracle Syntax is similar but slightly different.
- Oracle has queury optimization techniques in addition to triggers, SQL only uses triggers.
- Oracle executes statements in parallel, SQL executes commands serially.

Amazon S3 logging (323)

https://aws.amazon.com/premiumsupport/knowledge-center/advanced-audit-rds-mysql-cloudwatch/
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.MySQL.Options.AuditPlugin.html

- MariaDB Audit Plugin can capture events, operations, times, and objects affected by changes.
- Supports RDS, MySQL 8.0.25 and higher 8.0 versions
- Located at /rdsdbdata/log/audit
- Can configure MariaDB to export logs to Cloudwatch.

mySQL (324)

https://www.educba.com/mysql-commands/ mysql commands

https://kinsta.com/knowledgebase/what-is-mysql/
https://www.tutorialspoint.com/mysql/mysql-introduction.htm

- Relational Database (multiple seperate storage areas in tables).
- Client-server model, data resides in the server, client requests it.
- Works with many languages, C, C++, JAVA, etc.
- Standard SQL language.
- Open source

PostgreSQL (327)

https://www.postgresql.org/about/
https://hevodata.com/learn/postgresql-vs-sql-server/

- Open source, runs on all major operating systems.
- Can define own data types, build custom functions, write code from different languages.
- SQL standard is a base, but some syntax may be different.
- Supports Python, C, C++, Java, Javascript, Perl
- Provides encryption at various levels.

Encrypt Passwords (337)

https://www.geeksforgeeks.org/mysql-password-function/

- ENCRYPT PASSWORDS WITH PASSWORD(YOURSTRINGHERE)
- Actually, this is a hash, and can't be decrypted, so only comparisons can be made to the hashed password.

SQL Injection (341)

https://www.w3schools.com/sql/sql_injection.asp
https://www.malwarebytes.com/sql-injection
https://portswigger.net/web-security/sql-injection

- If someone puts in the username field "kennynguyen OR 1=1" the SQL statement will be "SELECT UserId, Name,
- Can drop tables by putting "105; DROP TABLE Suppliers" which will make the query "SELECT * FROM Users WHE
- Parameterize SQL queries to avoid this (SQL engine checks each parameter and treats it literally, not part of the
- `txtNam = getRequestString("CustomerName");`
- `txtAdd = getRequestString("Address");`
- `txtCit = getRequestString("City");`
- `txtSQL = "INSERT INTO Customers (CustomerName,Address,City) Values(@0,@1,@2)";`
- `db.Execute(txtSQL,txtNam,txtAdd,txtCit);`

SQL language (368)

https://www.w3schools.com/sql/default.asp
https://www.tutorialspoint.com/sql/sql-overview.htm

- Contains SQL syntax and functions

Database table Design (370)

https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5

https://medium.com/quick-code/10-best-database-design-practices-1f10f3441730

https://www.conceptatech.com/blog/best-practices-how-to-design-a-database

https://www.databasestar.com/entity-relationship-diagram/

- Normalize data to minimize redundancy
- Define constraints (foreign key, not null, etc)
- Database should be on a different server than the web server.
- Keep documentation on the design

AWS IAM (388)

https://aws.amazon.com/iam/

- Can assign users to different levels of access to each AWS service.
- Access is an explicit allow

Encrypt Database (430)

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html

- Encrypts data at rest
- AES-256 encryption
- Transparent data encryption is also supported for Oracle and SQL Server DBs.
- AWS KMS Key is used to encrypt the resource (use amazon managed key or make one)
- Can enable it on the RDS console or use the command --storage-encrypted
- Not supported for db.t2.micro (free tier)

Best Database security Practices (431)

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.Security.html

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.SSL.html

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/DataDurability.html

https://phoenixnap.com/kb/database-security

- Create IAM users for everyone, even the root user.
- Grant minimum permissions for each IAM account.
- Run DB instance inside the virtual private cloud.
- Use Security groups to control what IP addresses or EC2 instances can connect to the database.
- SSL connect to the DB instance.
- Use the encryption at rest feature for RDS.

- Use AWS CloudTrail and monitor API and user activity logging.
- Regular updates, backups (updates handled by Amazon)