**Final Exam Study Guide**

Know what SQL stands for

Know what Relational Databases (RDBMS) are and what they do, and that they contain a catalog that describes objects

Know the difference between the Database, Schema and a Table

Know the different between DML and DDL

Why use SQL?

* Allows users to access data in relational database management systems.
* Allows users to describe the data.
* Allows users to define the data in database and manipulate that data.
* Allows to embed within other languages using SQL modules, libraries & pre-compilers.
* Allows users to create and drop databases and tables.
* Allows users to create view, stored procedure, functions in a database.
* Allows users to set permissions on tables, procedures and views

Know about the (2) different types of Authentication (Mixed and Windows Authentication)

Know the difference between logins and users

Two separate security accounts exist in SQL Server, rather than the single username in oracle:

1. Logins – A login account is for an entire instance.

2. Users – A user account is for a particular database within an instance

Accounts must have **both** a login account for the instance and a user account for a database or they will not be able to connect.

In addition there are Windows accounts, typically handled through multi-server domains and maintained by system administrators rather than DBAs.

There are two methods of authentication for logins: Windows authentication (equivalent to an ops$ login in oracle) and mixed mode authentication, which is a username/password separate from the operating system.

Roles exist in SQL Server as in oracle. Roles are either Server level or Database level.

There are 7 fixed server level roles (sysadmin; dbcreator; diskadmin; processadmin; serveradmin; setupadmin; securityadmin, bulkadmin), 9 Fixed Database roles, and as many User-defined Database roles as you want to create.

The sysadmin server level role has DBA+ privileges, and so should not be granted outside the DBA team. Dbcreator is occasionally given to non-DBA users. The other fixed server level roles are in practice never used.

Three of the database level roles are commonly used:

db\_datareader – select any table

db\_datawriter – insert/update/delete any table

db\_owner – full power in that database, like an oracle schema owner

Object Permissions are granted to user accounts or database roles through the SQL Server Management Studio GUI. As in oracle, permissions can be implemented at column level. Roles can be assigned passwords.

Three separate object permissions exist in SQL Server:

1. **Grant** – can perform action

2. **Deny** – cannot perform action (strong). This applies even if the user account is a member of a role which has been granted the permission.

3. **Revoke** – cannot perform action (weak). This will be overridden by a grant to a role which the user account is a member of.

So Grant and Revoke are the same as in oracle. The additional Deny command is an extra strong form of Revoke.

Users can be granted the db\_owner role – but there is also a “dbo” user account for each database. This is associated with one particular login, the “database owner”. This can be seen in database properties in the “General” tab, and can be easily changed in (counterintuitively) the “Files” tab also in database properties. It can be good practice to enforce use of a single login for database owner, either a secure domain account or “sa”.

Know about SQL constraints

* NOT NULL Constraint: Ensures that a column cannot have NULL value.
* DEFAULT Constraint: Provides a default value for a column when none is specified.
* UNIQUE Constraint: Ensures that all values in a column are different.
* PRIMARY Key: Uniquely identified each rows/records in a database table.
* FOREIGN Key: Uniquely identified a rows/records in any another database table.
* CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
* INDEX: Use to create and retrieve data from the database very quickly.

Know what a Table is, and that is consists of rows and columns

Know what an Index is

Know what a View is

Views, which are kind of virtual tables, allow users to do the following:

* Structure data in a way that users or classes of users find natural or intuitive.
* Restrict access to the data such that a user can see and (sometimes) modify exactly what they need and no more.
* Summarize data from various tables which can be used to generate reports.

Know the following categories of data integrity that exist within the RDBMS

* **Entity Integrity:** There are no duplicate rows in a table.
* **Domain Integrity**: Enforces valid entries for a given column by restricting the type, the format, or the range of values.
* **Referential Integrity**: Rows cannot be deleted which are used by other records.
* **User-Defined Integrity**: Enforces some specific business rules that do not fall into entity, domain, or referential integrity.

Know the different levels of Normalization

Know the reason for using different data types

Know the different types of functions

* Date Functions
* Aggregate Functions
* String functions

Backup can be done through third party backup agents for SQL Server. But the preferred backup methodology is to use SQL Server‟s own backup utility. This will dump backup files to disk, which will then be backed up by the filesystem backup to tape. The option to delete old backups should be ticked. Backups should be kept for e.g. 4 weeks for small databases, with just a single backup kept on disk for large databases. The disk backup files can still be retrieved as operating system files from tape backup if required.

Backups can be written direct from SQL Server to

* disk file
* tape
* pipe

Most commonly SQL Server uses backups to disk file.

All SQL Server backups are online (hot). While a backup is in progress, it is impossible to:

* create or modify databases;
* autogrow files;
* create indexes;
* perform non-logged operations.

There are three kinds of SQL Server backup:

1. Full backup – backs up data files and the part of the transaction log that took place during the full backup. Equivalent to a full online backup in oracle.

2. Differential backup – backs up the extents which have been modified since the last full backup, and the part of the transaction log that took place during the differential backup.

3. Transaction log backup with “truncate the transaction log” option – backs up and then truncates the transaction log. Equivalent to a log switch in an oracle instance in archivelog mode.

4. Transaction log backup with “backup the tail of the log” option – backs up the transaction log without truncating. This would be run immediately before attempting a restore in order to keep a copy of the current transaction log.