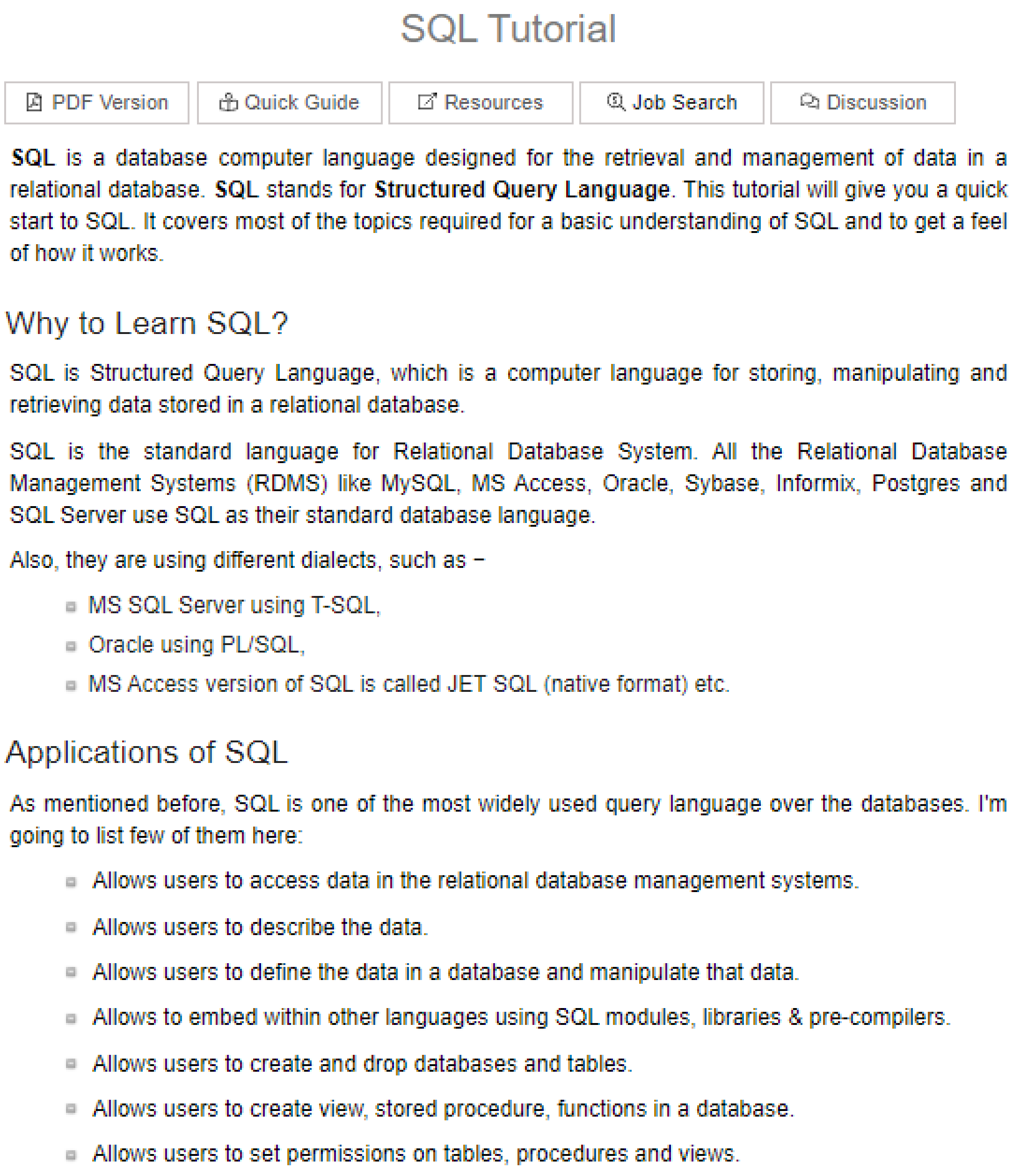
Solve 4 Questions out of 6

Database Outlines

1. SQL



SQL Tutorial

SQL is a database computer language designed for the retrieval and management of data in a relational database. **SQL** stands for **Structured Query Language**. This tutorial will give you a quick start to SQL. It covers most of the topics required for a basic understanding of SQL and to get a feel of how it works.

Why to Learn SQL?

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

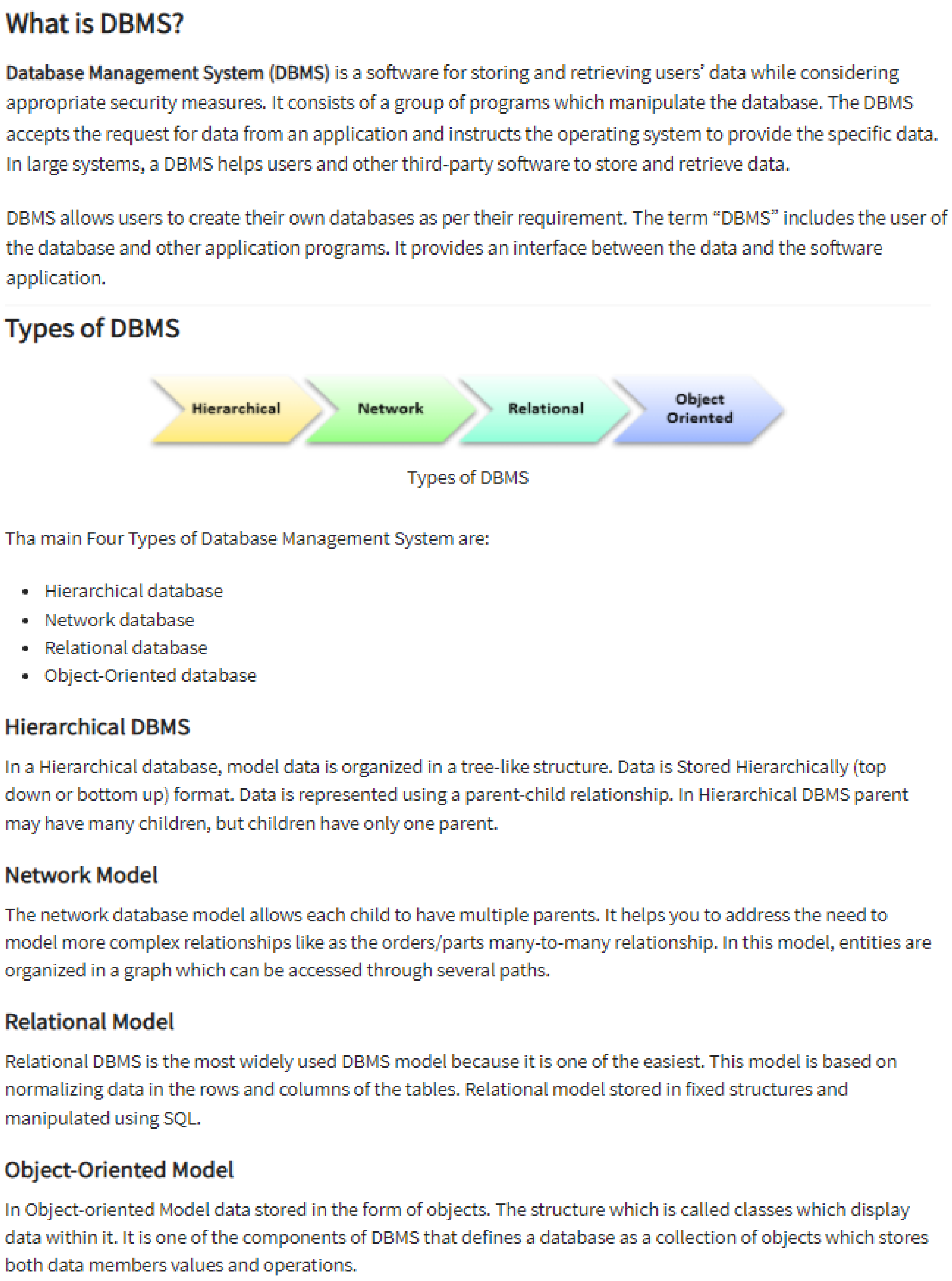
Applications of SQL

As mentioned before, SQL is one of the most widely used query language over the databases. I'm going to list few of them here:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a 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.

2. DBMS, Types of DBMS

Solve 4 Questions out of 6



What is DBMS?

Database Management System (DBMS) is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

DBMS allows users to create their own databases as per their requirement. The term "DBMS" includes the user of the database and other application programs. It provides an interface between the data and the software application.

Types of DBMS

Types of DBMS

Tha main Four Types of Database Management System are:

- Hierarchical database
- Network database
- Relational database
- Object-Oriented database

Hierarchical DBMS

In a Hierarchical database, model data is organized in a tree-like structure. Data is Stored Hierarchically (top down or bottom up) format. Data is represented using a parent-child relationship. In Hierarchical DBMS parent may have many children, but children have only one parent.

Network Model

The network database model allows each child to have multiple parents. It helps you to address the need to model more complex relationships like as the orders/parts many-to-many relationship. In this model, entities are organized in a graph which can be accessed through several paths.

Relational Model

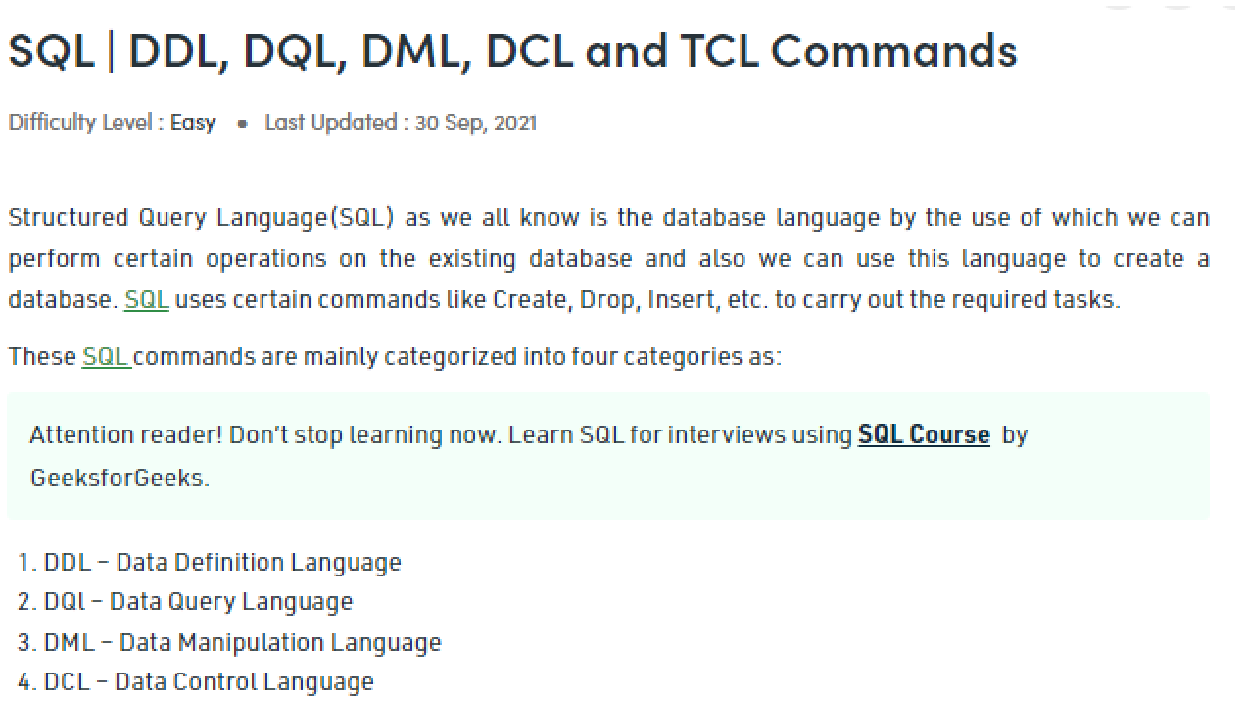
Relational DBMS is the most widely used DBMS model because it is one of the easiest. This model is based on normalizing data in the rows and columns of the tables. Relational model stored in fixed structures and manipulated using SQL.

Object-Oriented Model

In Object-oriented Model data stored in the form of objects. The structure which is called classes which display data within it. It is one of the components of DBMS that defines a database as a collection of objects which stores both data members values and operations.

Solve 4 Questions out of 6

3. Sub languages of SQL



SQL | DDL, DQL, DML, DCL and TCL Commands

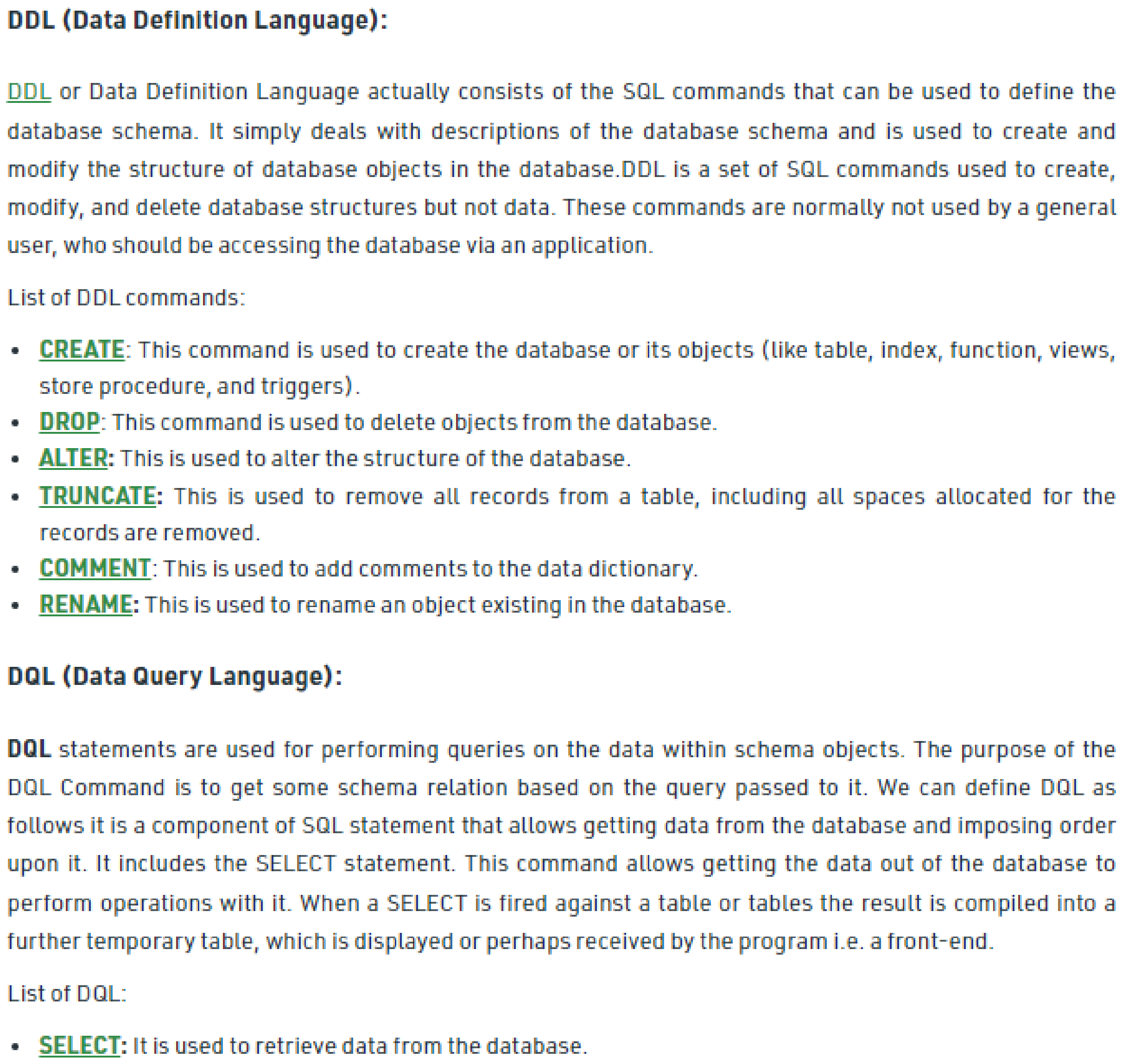
Difficulty Level : Easy • Last Updated : 30 Sep, 2021

Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

Solve 4 Questions out of 6



DDL (Data Definition Language):

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.DDL is a set of SQL commands used to create, modify, and delete database structures but not data. These commands are normally not used by a general user, who should be accessing the database via an application.

List of DDL commands:

- **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP**: This command is used to delete objects from the database.
- **ALTER:** This is used to alter the structure of the database.
- **TRUNCATE:** This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT**: This is used to add comments to the data dictionary.
- **RENAME:** This is used to rename an object existing in the database.

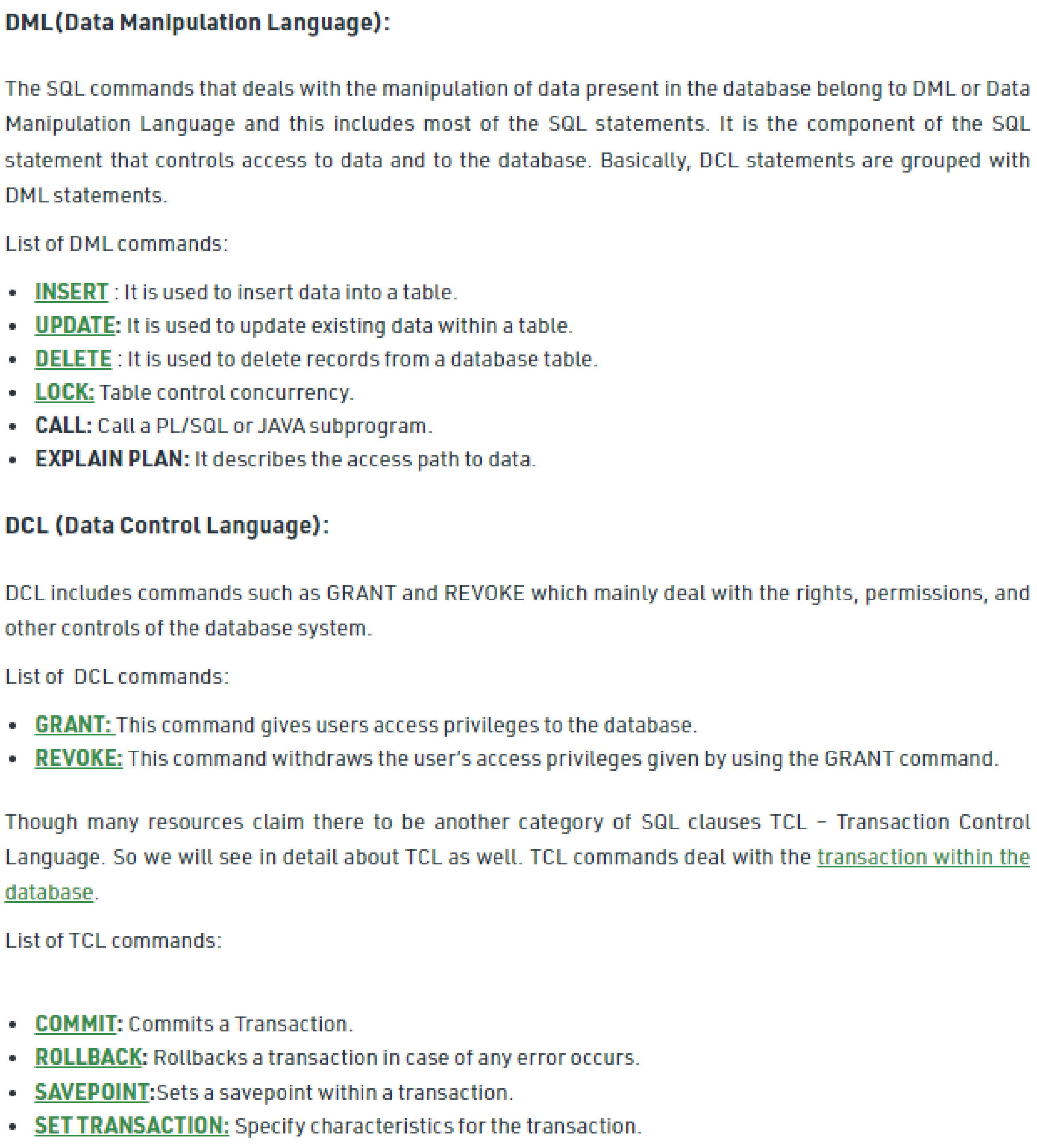
DQL (Data Query Language):

DQL statements are used for performing queries on the data within schema objects. The purpose of the DQL Command is to get some schema relation based on the query passed to it. We can define DQL as follows it is a component of SQL statement that allows getting data from the database and imposing order upon it. It includes the SELECT statement. This command allows getting the data out of the database to perform operations with it. When a SELECT is fired against a table or tables the result is compiled into a further temporary table, which is displayed or perhaps received by the program i.e. a front-end.

List of DQL:

- **SELECT:** It is used to retrieve data from the database.

Solve 4 Questions out of 6



DML(Data Manipulation Language):

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

List of DML commands:

- **INSERT** : It is used to insert data into a table.
- **UPDATE:** It is used to update existing data within a table.
- **DELETE** : It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

DCL (Data Control Language):

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

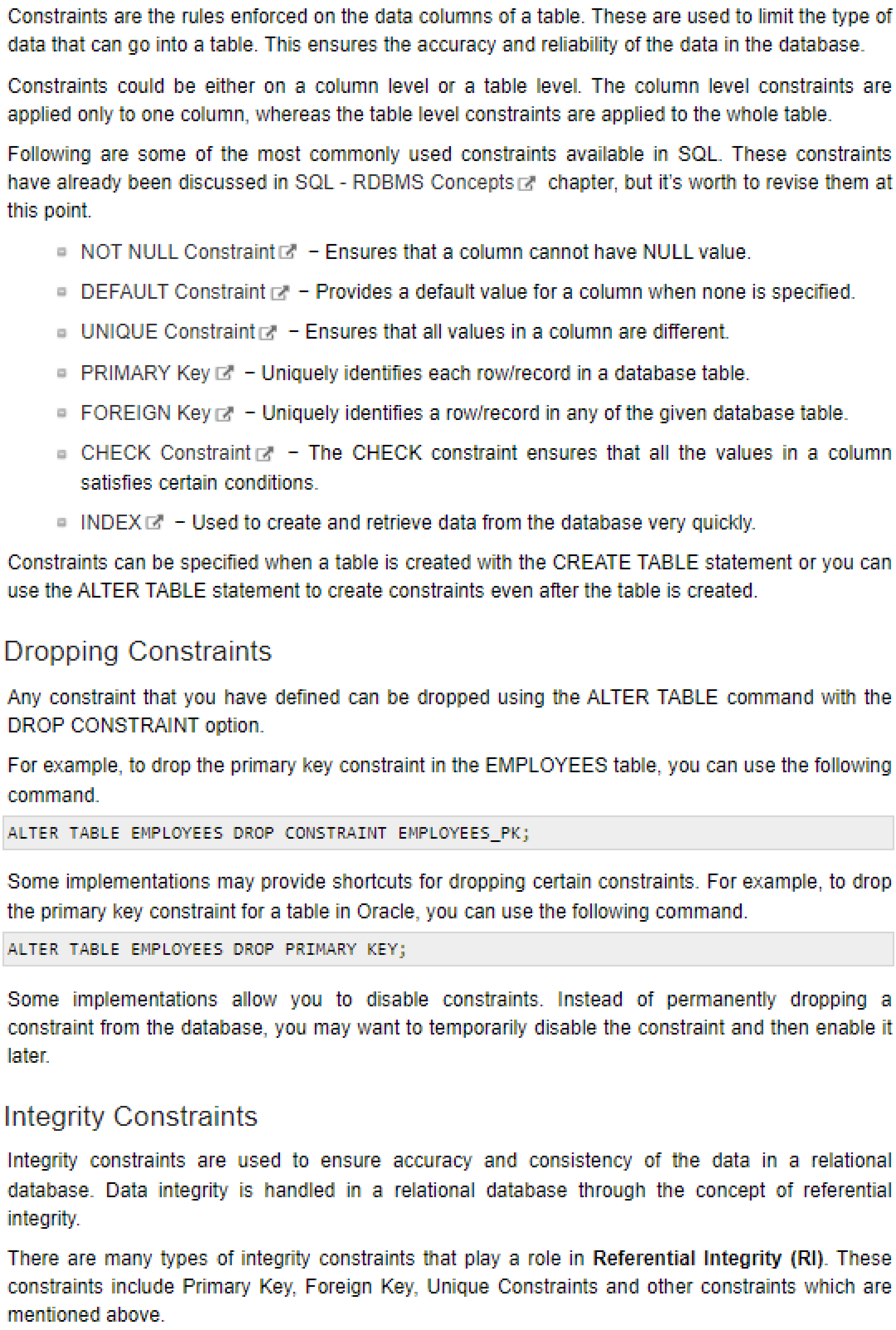
Though many resources claim there to be another category of SQL clauses TCL – Transaction Control Language. So we will see in detail about TCL as well. TCL commands deal with the transaction within the database.

List of TCL commands:

- **COMMIT:** Commits a Transaction.
- **ROLLBACK:** Rollbacks a transaction in case of any error occurs.
- **SAVEPOINT:**Sets a savepoint within a transaction.
- **SET TRANSACTION:** Specify characteristics for the transaction.

4. Table Constraints

Solve 4 Questions out of 6



Constraints are the rules enforced on the data columns of a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints could be either on a column level or a table level. The column level constraints are applied only to one column, whereas the table level constraints are applied to the whole table.

Following are some of the most commonly used constraints available in SQL. These constraints have already been discussed in SQL - RDBMS Concepts chapter, but it's worth to revise them at this point.

- 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 identifies each row/record in a database table.
- FOREIGN Key − Uniquely identifies a row/record in any of the given database table.
- CHECK Constraint − The CHECK constraint ensures that all the values in a column satisfies certain conditions.
- INDEX − Used to create and retrieve data from the database very quickly.

Constraints can be specified when a table is created with the CREATE TABLE statement or you can use the ALTER TABLE statement to create constraints even after the table is created.

Dropping Constraints

Any constraint that you have defined can be dropped using the ALTER TABLE command with the DROP CONSTRAINT option.

For example, to drop the primary key constraint in the EMPLOYEES table, you can use the following command.

```
ALTER TABLE EMPLOYEES DROP CONSTRAINT EMPLOYEES_PK;
```

Some implementations may provide shortcuts for dropping certain constraints. For example, to drop the primary key constraint for a table in Oracle, you can use the following command.

```
ALTER TABLE EMPLOYEES DROP PRIMARY KEY;
```

Some implementations allow you to disable constraints. Instead of permanently dropping a constraint from the database, you may want to temporarily disable the constraint and then enable it later.

Integrity Constraints

Integrity constraints are used to ensure accuracy and consistency of the data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity.

There are many types of integrity constraints that play a role in **Referential Integrity (RI)**. These constraints include Primary Key, Foreign Key, Unique Constraints and other constraints which are mentioned above.

Solve 4 Questions out of 6

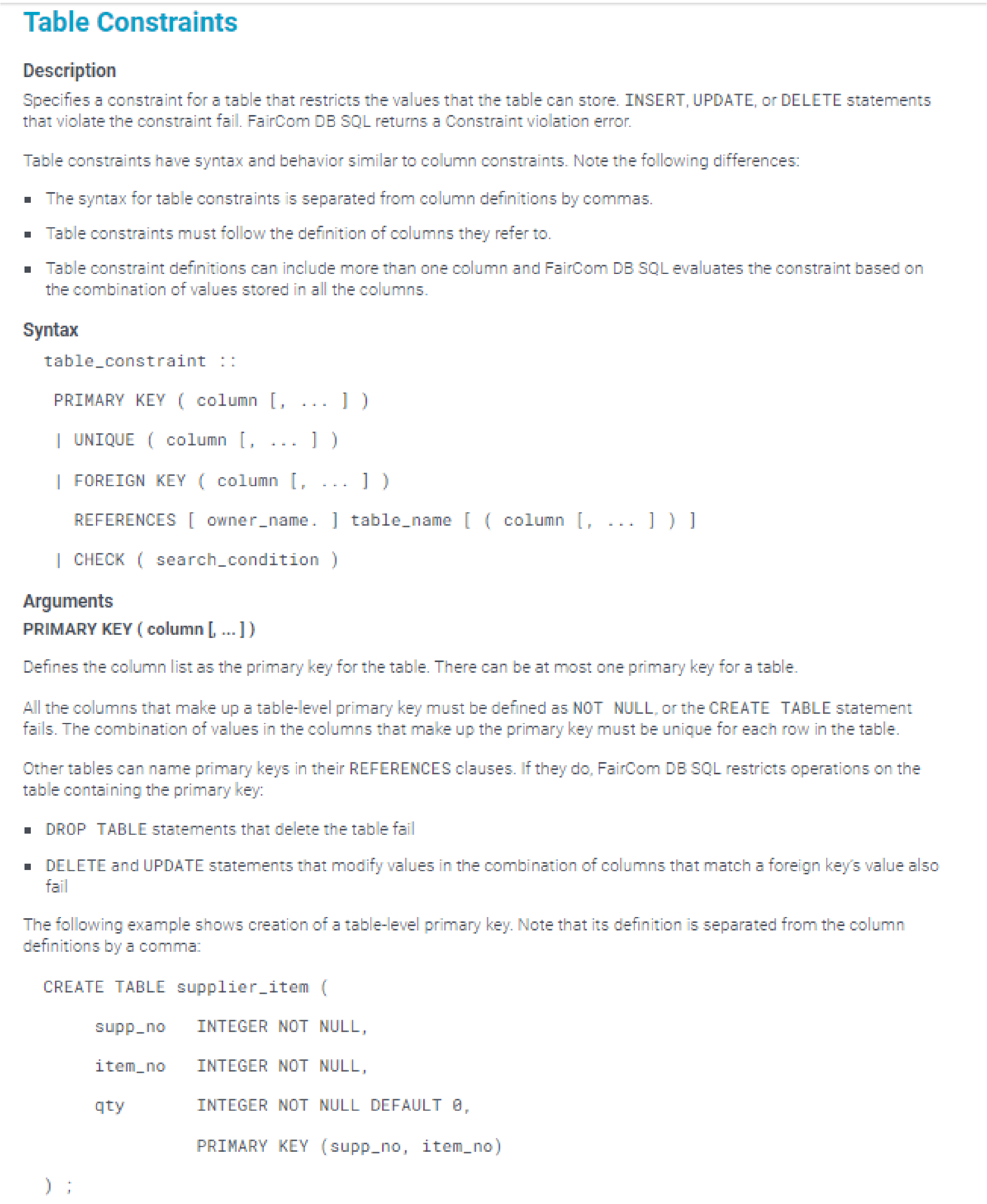


Table Constraints

Description

Specifies a constraint for a table that restricts the values that the table can store. INSERT, UPDATE, or DELETE statements that violate the constraint fail. FairCom DB SQL returns a Constraint violation error.

Table constraints have syntax and behavior similar to column constraints. Note the following differences:

- The syntax for table constraints is separated from column definitions by commas.
- Table constraints must follow the definition of columns they refer to.
- Table constraint definitions can include more than one column and FairCom DB SQL evaluates the constraint based on the combination of values stored in all the columns.

Syntax

```
table_constraint ::

 PRIMARY KEY ( column [, ... ] )

 | UNIQUE ( column [, ... ] )

 | FOREIGN KEY ( column [, ... ] )

   REFERENCES [ owner_name. ] table_name [ ( column [, ... ] ) ]

 | CHECK ( search_condition )
```

Arguments

PRIMARY KEY (column [, ...])

Defines the column list as the primary key for the table. There can be at most one primary key for a table.

All the columns that make up a table-level primary key must be defined as NOT NULL, or the CREATE TABLE statement fails. The combination of values in the columns that make up the primary key must be unique for each row in the table.

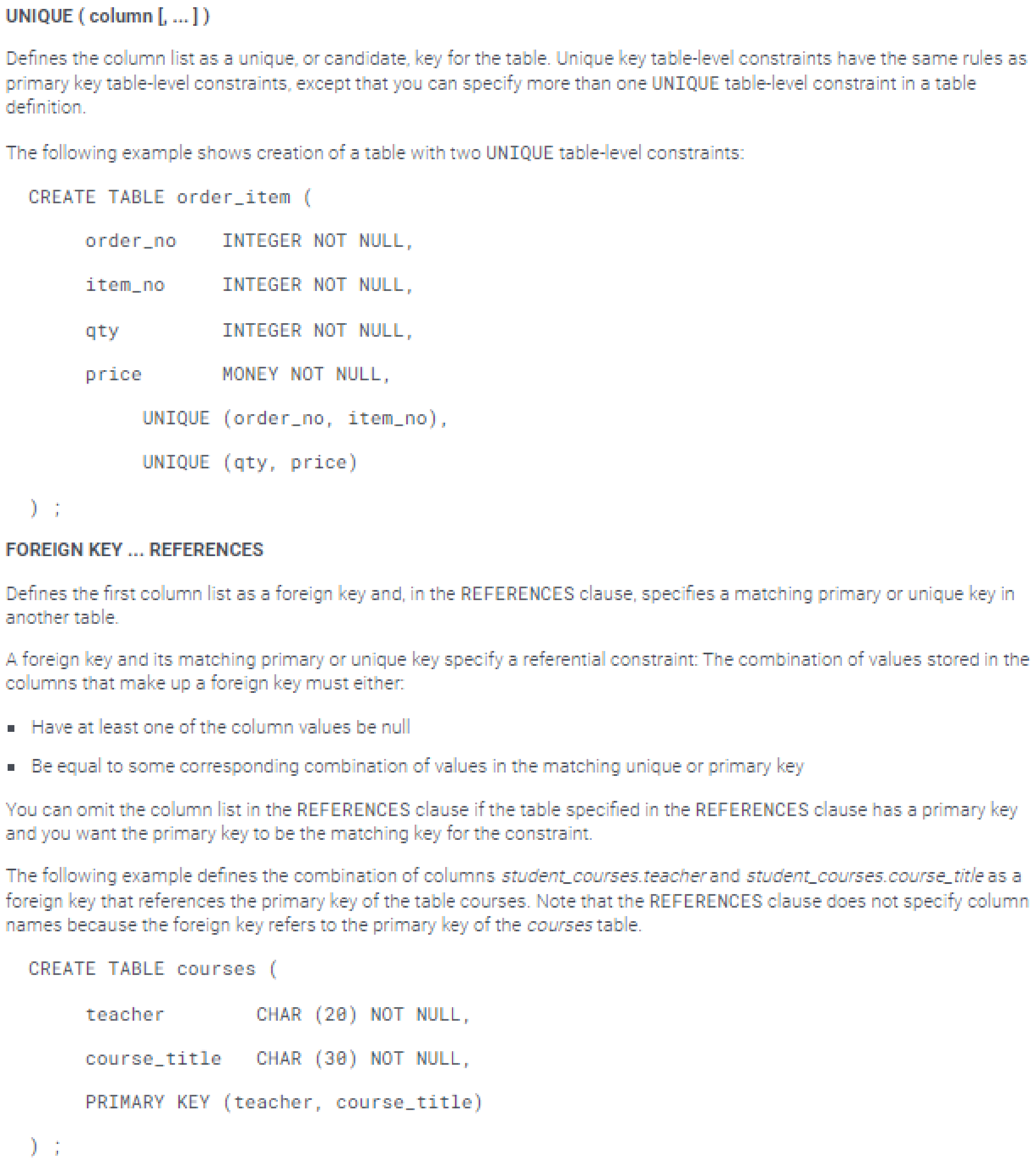
Other tables can name primary keys in their REFERENCES clauses. If they do, FairCom DB SQL restricts operations on the table containing the primary key:

- DROP TABLE statements that delete the table fail
- DELETE and UPDATE statements that modify values in the combination of columns that match a foreign key's value also fail

The following example shows creation of a table-level primary key. Note that its definition is separated from the column definitions by a comma:

```
CREATE TABLE supplier_item (

     supp_no    INTEGER NOT NULL,

     item_no    INTEGER NOT NULL,

     qty        INTEGER NOT NULL DEFAULT 0,

                PRIMARY KEY (supp_no, item_no)

) ;
```

Solve 4 Questions out of 6



UNIQUE (column [, ...])

Defines the column list as a unique, or candidate, key for the table. Unique key table-level constraints have the same rules as primary key table-level constraints, except that you can specify more than one UNIQUE table-level constraint in a table definition.

The following example shows creation of a table with two UNIQUE table-level constraints:

```
CREATE TABLE order_item (

     order_no    INTEGER NOT NULL,

     item_no     INTEGER NOT NULL,

     qty         INTEGER NOT NULL,

     price       MONEY NOT NULL,

          UNIQUE (order_no, item_no),

          UNIQUE (qty, price)

) ;
```

FOREIGN KEY ... REFERENCES

Defines the first column list as a foreign key and, in the REFERENCES clause, specifies a matching primary or unique key in another table.

A foreign key and its matching primary or unique key specify a referential constraint: The combination of values stored in the columns that make up a foreign key must either:

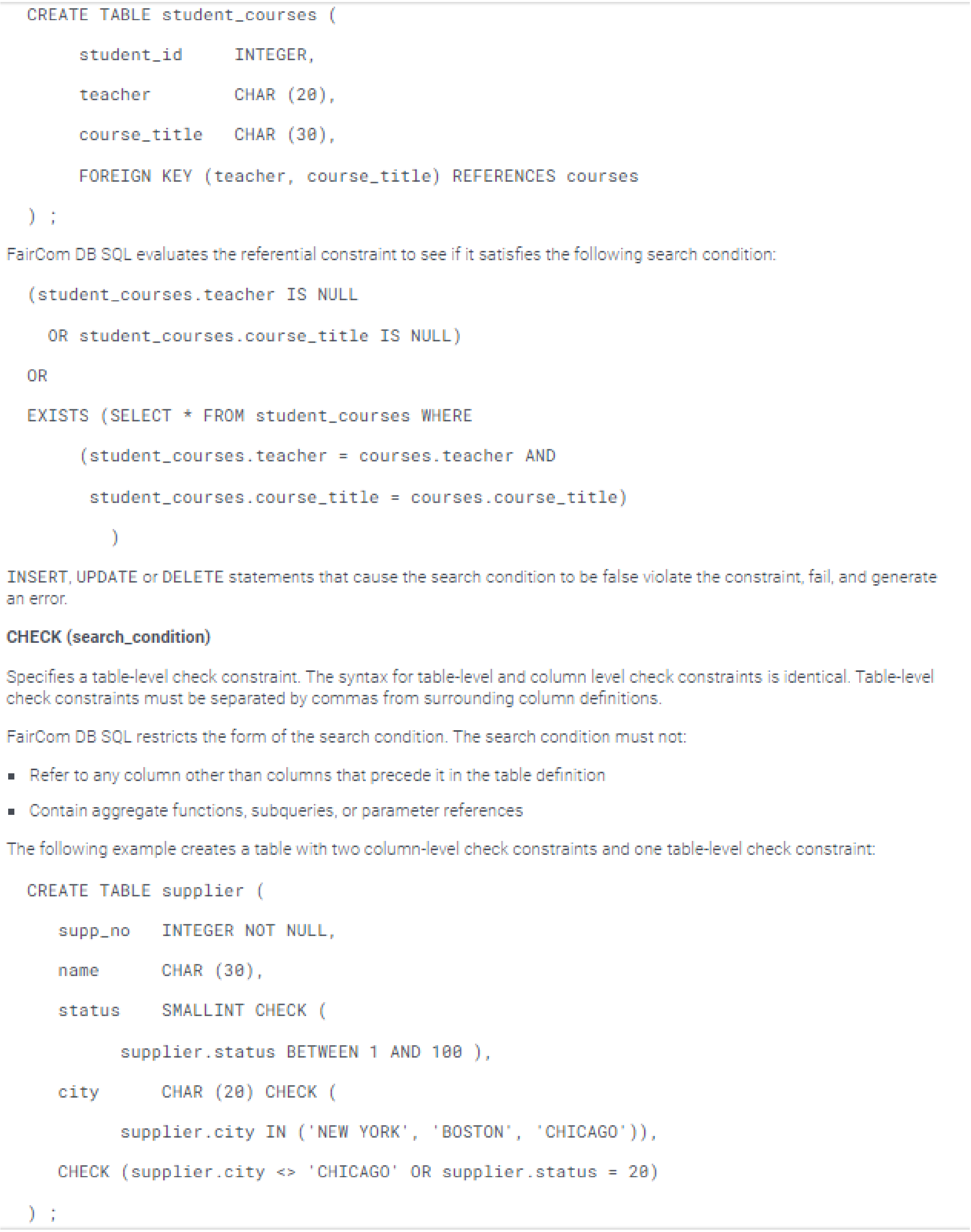
- Have at least one of the column values be null
- Be equal to some corresponding combination of values in the matching unique or primary key

You can omit the column list in the REFERENCES clause if the table specified in the REFERENCES clause has a primary key and you want the primary key to be the matching key for the constraint.

The following example defines the combination of columns *student_courses.teacher* and *student_courses.course_title* as a foreign key that references the primary key of the table courses. Note that the REFERENCES clause does not specify column names because the foreign key refers to the primary key of the *courses* table.

```
CREATE TABLE courses (

     teacher        CHAR (20) NOT NULL,

     course_title   CHAR (30) NOT NULL,

     PRIMARY KEY (teacher, course_title)

) ;
```

Solve 4 Questions out of 6



```
CREATE TABLE student_courses (
     student_id     INTEGER,
     teacher        CHAR (20),
     course_title   CHAR (30),
     FOREIGN KEY (teacher, course_title) REFERENCES courses
) ;
```

FairCom DB SQL evaluates the referential constraint to see if it satisfies the following search condition:

```
(student_courses.teacher IS NULL
  OR student_courses.course_title IS NULL)
OR
EXISTS (SELECT * FROM student_courses WHERE
     (student_courses.teacher = courses.teacher AND
      student_courses.course_title = courses.course_title)
        )
```

INSERT, UPDATE or DELETE statements that cause the search condition to be false violate the constraint, fail, and generate an error.

CHECK (search_condition)

Specifies a table-level check constraint. The syntax for table-level and column level check constraints is identical. Table-level check constraints must be separated by commas from surrounding column definitions.

FairCom DB SQL restricts the form of the search condition. The search condition must not:

- Refer to any column other than columns that precede it in the table definition
- Contain aggregate functions, subqueries, or parameter references

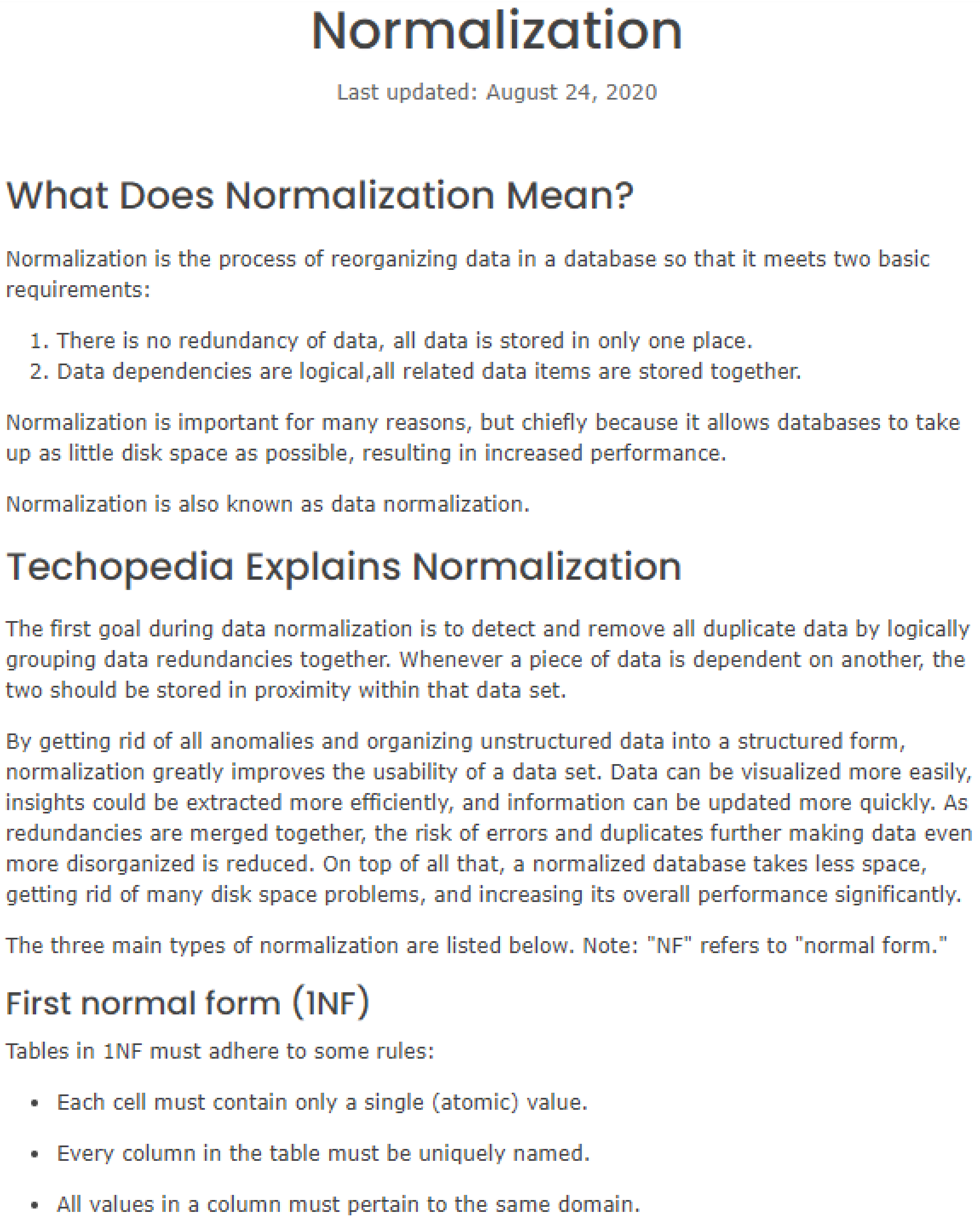
The following example creates a table with two column-level check constraints and one table-level check constraint:

```
CREATE TABLE supplier (
   supp_no   INTEGER NOT NULL,
   name      CHAR (30),
   status    SMALLINT CHECK (
         supplier.status BETWEEN 1 AND 100 ),
   city      CHAR (20) CHECK (
         supplier.city IN ('NEW YORK', 'BOSTON', 'CHICAGO')),
   CHECK (supplier.city <> 'CHICAGO' OR supplier.status = 20)
) ;
```

Solve 4 Questions out of 6

5. Oracle Instance

6. Data Normalization



Normalization

Last updated: August 24, 2020

What Does Normalization Mean?

Normalization is the process of reorganizing data in a database so that it meets two basic requirements:

1. There is no redundancy of data, all data is stored in only one place.
2. Data dependencies are logical,all related data items are stored together.

Normalization is important for many reasons, but chiefly because it allows databases to take up as little disk space as possible, resulting in increased performance.

Normalization is also known as data normalization.

Techopedia Explains Normalization

The first goal during data normalization is to detect and remove all duplicate data by logically grouping data redundancies together. Whenever a piece of data is dependent on another, the two should be stored in proximity within that data set.

By getting rid of all anomalies and organizing unstructured data into a structured form, normalization greatly improves the usability of a data set. Data can be visualized more easily, insights could be extracted more efficiently, and information can be updated more quickly. As redundancies are merged together, the risk of errors and duplicates further making data even more disorganized is reduced. On top of all that, a normalized database takes less space, getting rid of many disk space problems, and increasing its overall performance significantly.

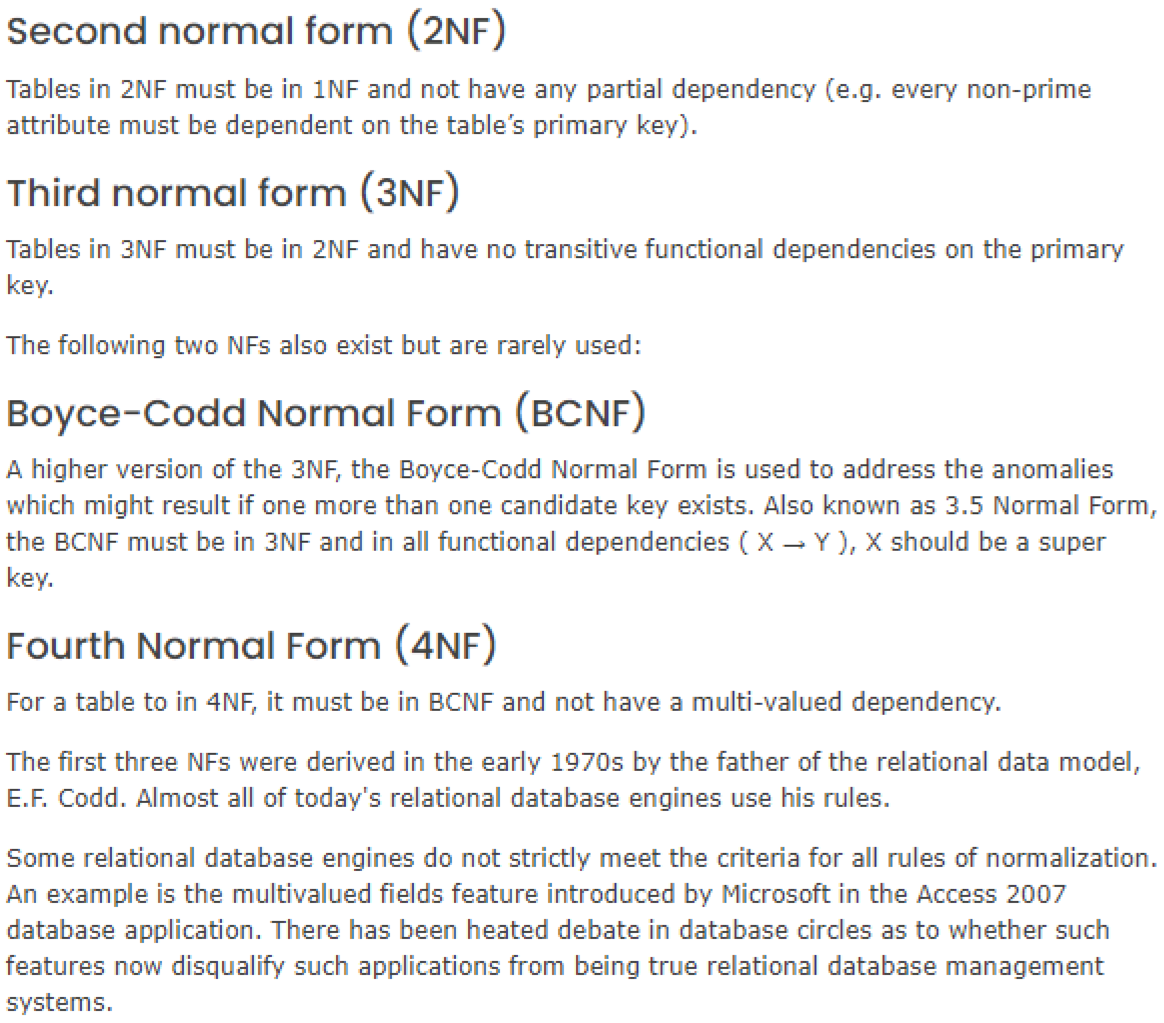
The three main types of normalization are listed below. Note: "NF" refers to "normal form."

First normal form (1NF)

Tables in 1NF must adhere to some rules:

- Each cell must contain only a single (atomic) value.
- Every column in the table must be uniquely named.
- All values in a column must pertain to the same domain.

Solve 4 Questions out of 6



Second normal form (2NF)

Tables in 2NF must be in 1NF and not have any partial dependency (e.g. every non-prime attribute must be dependent on the table’s primary key).

Third normal form (3NF)

Tables in 3NF must be in 2NF and have no transitive functional dependencies on the primary key.

The following two NFs also exist but are rarely used:

Boyce-Codd Normal Form (BCNF)

A higher version of the 3NF, the Boyce-Codd Normal Form is used to address the anomalies which might result if one more than one candidate key exists. Also known as 3.5 Normal Form, the BCNF must be in 3NF and in all functional dependencies (X → Y), X should be a super key.

Fourth Normal Form (4NF)

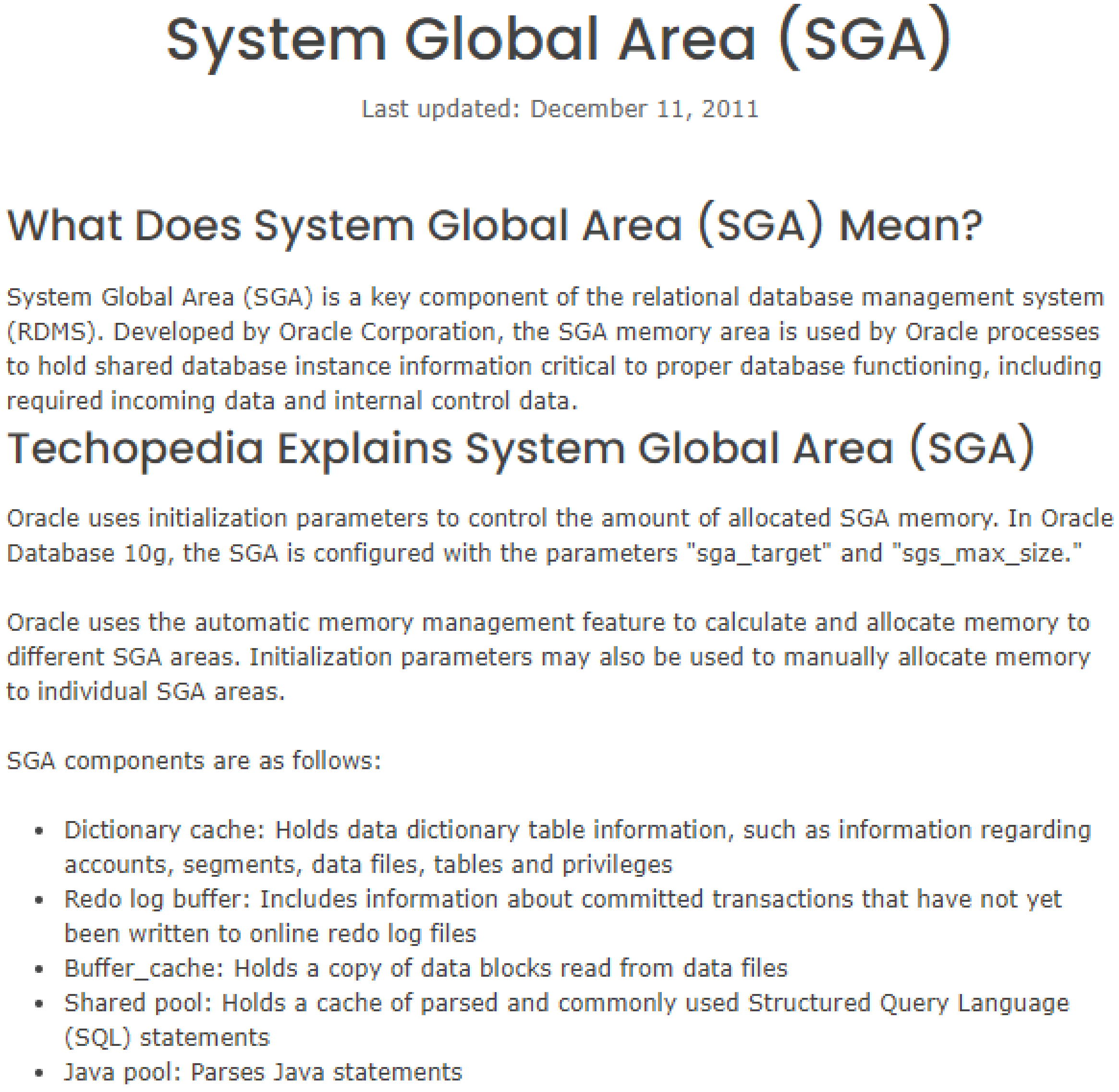
For a table to in 4NF, it must be in BCNF and not have a multi-valued dependency.

The first three NFs were derived in the early 1970s by the father of the relational data model, E.F. Codd. Almost all of today's relational database engines use his rules.

Some relational database engines do not strictly meet the criteria for all rules of normalization. An example is the multivalued fields feature introduced by Microsoft in the Access 2007 database application. There has been heated debate in database circles as to whether such features now disqualify such applications from being true relational database management systems.

Solve 4 Questions out of 6

7. Systems Global Area (SGA)/Oracle background processes



System Global Area (SGA)

Last updated: December 11, 2011

What Does System Global Area (SGA) Mean?

System Global Area (SGA) is a key component of the relational database management system (RDMS). Developed by Oracle Corporation, the SGA memory area is used by Oracle processes to hold shared database instance information critical to proper database functioning, including required incoming data and internal control data.

Techopedia Explains System Global Area (SGA)

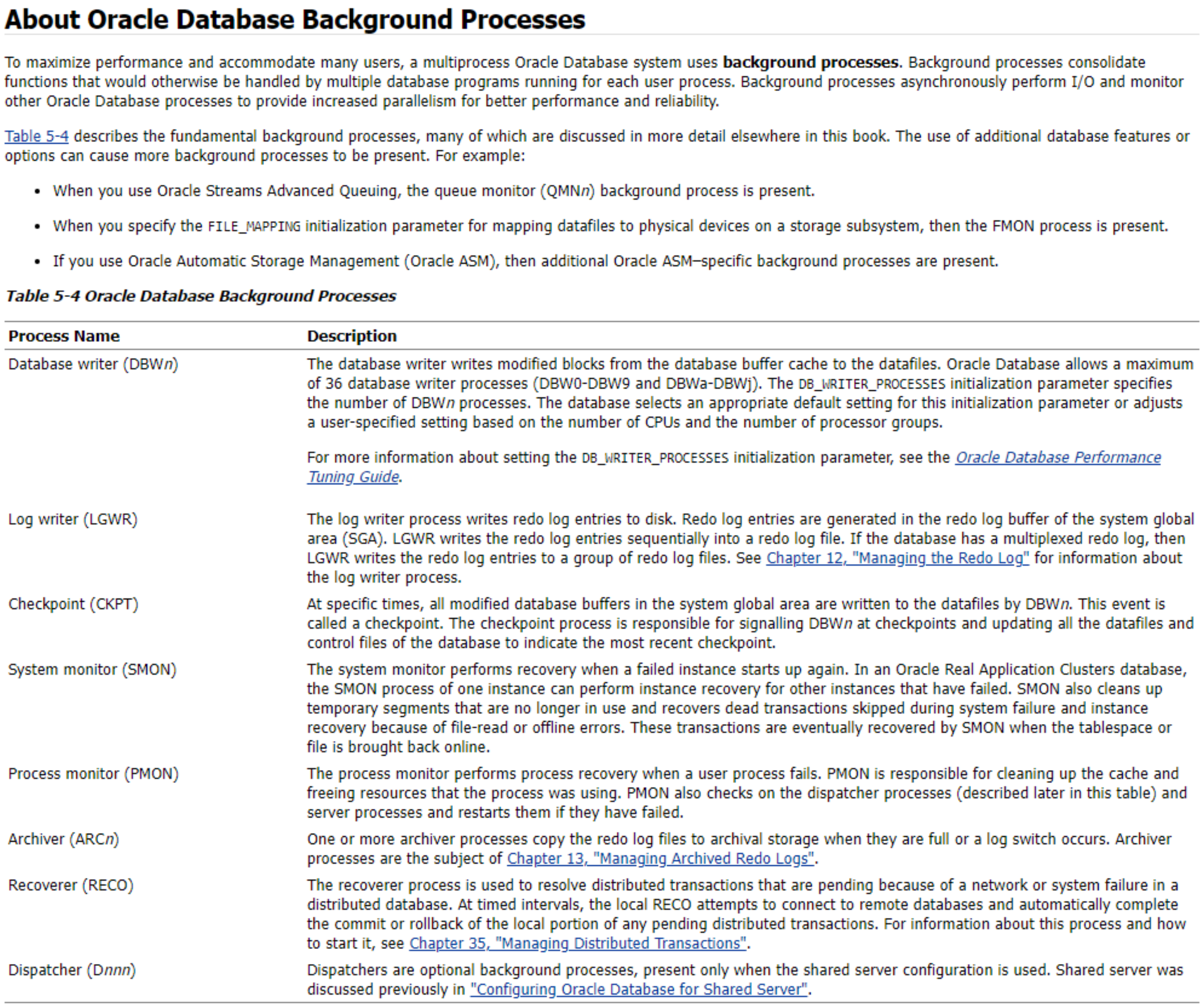
Oracle uses initialization parameters to control the amount of allocated SGA memory. In Oracle Database 10g, the SGA is configured with the parameters "sga_target" and "sgs_max_size."

Oracle uses the automatic memory management feature to calculate and allocate memory to different SGA areas. Initialization parameters may also be used to manually allocate memory to individual SGA areas.

SGA components are as follows:

- Dictionary cache: Holds data dictionary table information, such as information regarding accounts, segments, data files, tables and privileges
- Redo log buffer: Includes information about committed transactions that have not yet been written to online redo log files
- Buffer_cache: Holds a copy of data blocks read from data files
- Shared pool: Holds a cache of parsed and commonly used Structured Query Language (SQL) statements
- Java pool: Parses Java statements

Solve 4 Questions out of 6



About Oracle Database Background Processes

To maximize performance and accommodate many users, a multiprocess Oracle Database system uses **background processes**. Background processes consolidate functions that would otherwise be handled by multiple database programs running for each user process. Background processes asynchronously perform I/O and monitor other Oracle Database processes to provide increased parallelism for better performance and reliability.

Table 5-4 describes the fundamental background processes, many of which are discussed in more detail elsewhere in this book. The use of additional database features or options can cause more background processes to be present. For example:

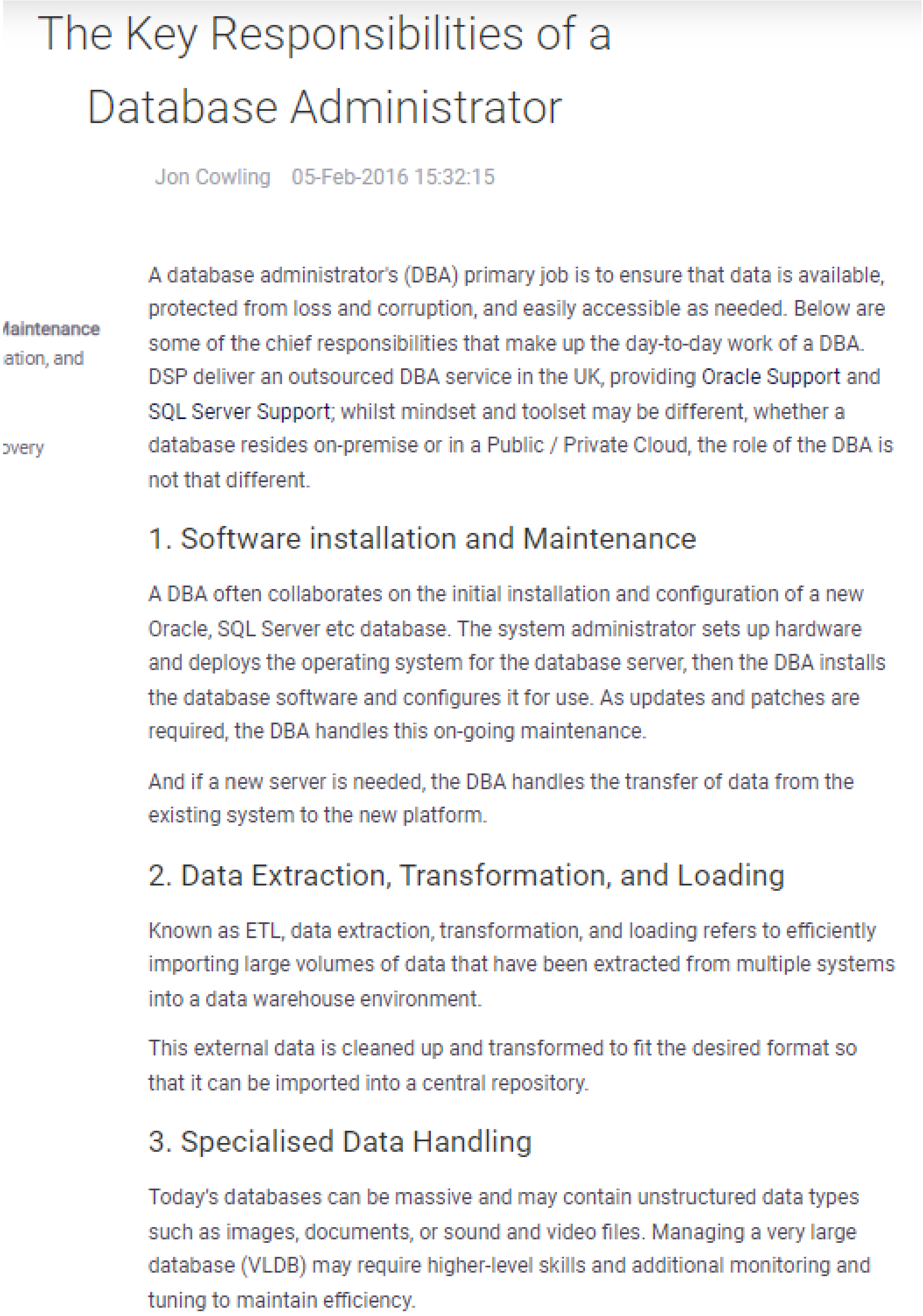
- When you use Oracle Streams Advanced Queuing, the queue monitor (QMN*n*) background process is present.
- When you specify the `FILE_MAPPING` initialization parameter for mapping datafiles to physical devices on a storage subsystem, then the FMON process is present.
- If you use Oracle Automatic Storage Management (Oracle ASM), then additional Oracle ASM–specific background processes are present.

Table 5-4 Oracle Database Background Processes

Process Name	**Description**
Database writer (DBW*n*)	The database writer writes modified blocks from the database buffer cache to the datafiles. Oracle Database allows a maximum of 36 database writer processes (DBW0-DBW9 and DBWa-DBWj). The `DB_WRITER_PROCESSES` initialization parameter specifies the number of DBW*n* processes. The database selects an appropriate default setting for this initialization parameter or adjusts a user-specified setting based on the number of CPUs and the number of processor groups. For more information about setting the `DB_WRITER_PROCESSES` initialization parameter, see the *Oracle Database Performance Tuning Guide*.
Log writer (LGWR)	The log writer process writes redo log entries to disk. Redo log entries are generated in the redo log buffer of the system global area (SGA). LGWR writes the redo log entries sequentially into a redo log file. If the database has a multiplexed redo log, then LGWR writes the redo log entries to a group of redo log files. See Chapter 12, "Managing the Redo Log" for information about the log writer process.
Checkpoint (CKPT)	At specific times, all modified database buffers in the system global area are written to the datafiles by DBW*n*. This event is called a checkpoint. The checkpoint process is responsible for signalling DBW*n* at checkpoints and updating all the datafiles and control files of the database to indicate the most recent checkpoint.
System monitor (SMON)	The system monitor performs recovery when a failed instance starts up again. In an Oracle Real Application Clusters database, the SMON process of one instance can perform instance recovery for other instances that have failed. SMON also cleans up temporary segments that are no longer in use and recovers dead transactions skipped during system failure and instance recovery because of file-read or offline errors. These transactions are eventually recovered by SMON when the tablespace or file is brought back online.
Process monitor (PMON)	The process monitor performs process recovery when a user process fails. PMON is responsible for cleaning up the cache and freeing resources that the process was using. PMON also checks on the dispatcher processes (described later in this table) and server processes and restarts them if they have failed.
Archiver (ARC*n*)	One or more archiver processes copy the redo log files to archival storage when they are full or a log switch occurs. Archiver processes are the subject of Chapter 13, "Managing Archived Redo Logs".
Recoverer (RECO)	The recoverer process is used to resolve distributed transactions that are pending because of a network or system failure in a distributed database. At timed intervals, the local RECO attempts to connect to remote databases and automatically complete the commit or rollback of the local portion of any pending distributed transactions. For information about this process and how to start it, see Chapter 35, "Managing Distributed Transactions".
Dispatcher (D*nnn*)	Dispatchers are optional background processes, present only when the shared server configuration is used. Shared server was discussed previously in "Configuring Oracle Database for Shared Server".

Solve 4 Questions out of 6

8. Responsibilities of Database Administrator (DBA)



The Key Responsibilities of a Database Administrator

Jon Cowling 05-Feb-2016 15:32:15

laintenance
ation, and

overy

A database administrator's (DBA) primary job is to ensure that data is available, protected from loss and corruption, and easily accessible as needed. Below are some of the chief responsibilities that make up the day-to-day work of a DBA. DSP deliver an outsourced DBA service in the UK, providing Oracle Support and SQL Server Support; whilst mindset and toolset may be different, whether a database resides on-premise or in a Public / Private Cloud, the role of the DBA is not that different.

1. Software installation and Maintenance

A DBA often collaborates on the initial installation and configuration of a new Oracle, SQL Server etc database. The system administrator sets up hardware and deploys the operating system for the database server, then the DBA installs the database software and configures it for use. As updates and patches are required, the DBA handles this on-going maintenance.

And if a new server is needed, the DBA handles the transfer of data from the existing system to the new platform.

2. Data Extraction, Transformation, and Loading

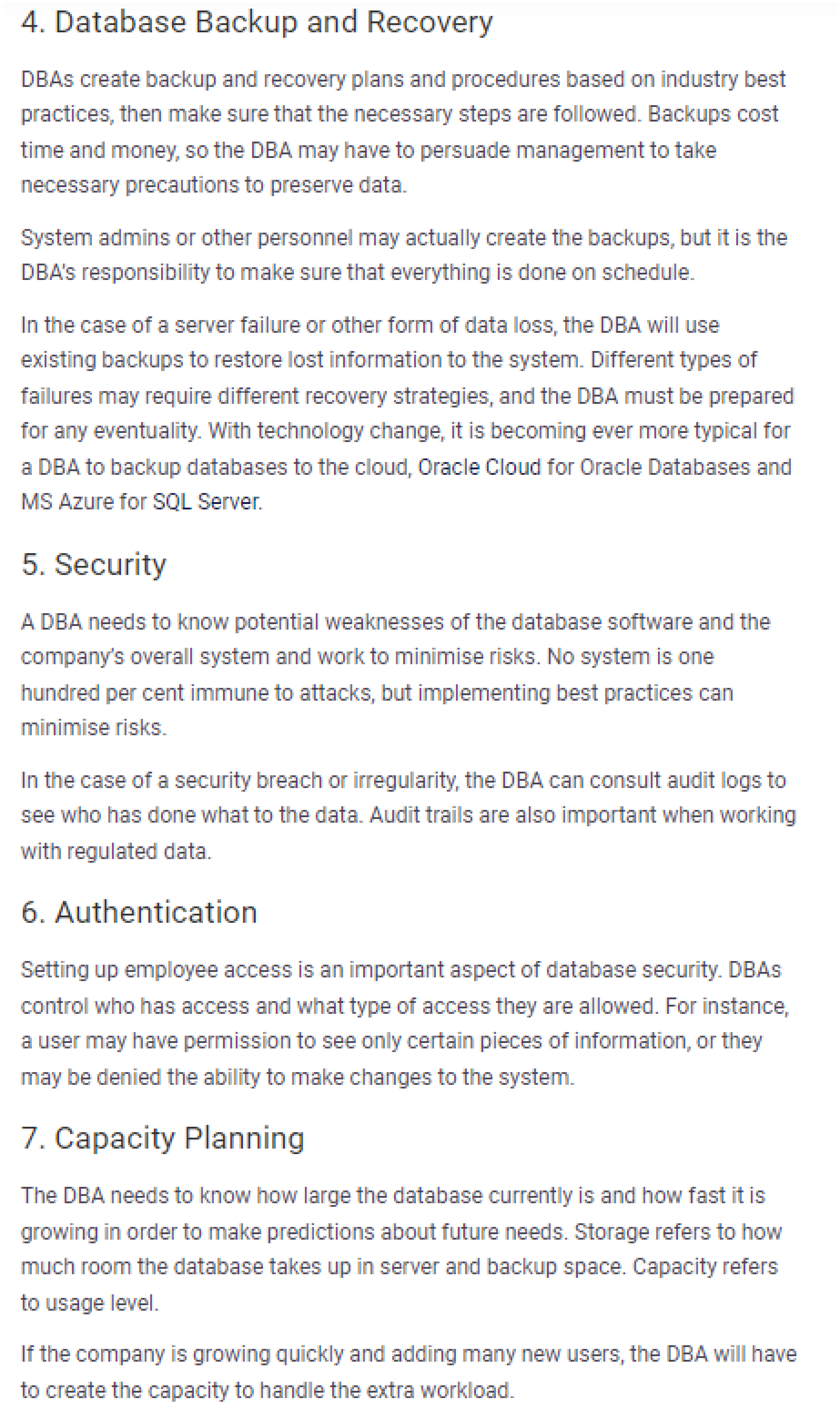
Known as ETL, data extraction, transformation, and loading refers to efficiently importing large volumes of data that have been extracted from multiple systems into a data warehouse environment.

This external data is cleaned up and transformed to fit the desired format so that it can be imported into a central repository.

3. Specialised Data Handling

Today's databases can be massive and may contain unstructured data types such as images, documents, or sound and video files. Managing a very large database (VLDB) may require higher-level skills and additional monitoring and tuning to maintain efficiency.

Solve 4 Questions out of 6



4. Database Backup and Recovery

DBAs create backup and recovery plans and procedures based on industry best practices, then make sure that the necessary steps are followed. Backups cost time and money, so the DBA may have to persuade management to take necessary precautions to preserve data.

System admins or other personnel may actually create the backups, but it is the DBA's responsibility to make sure that everything is done on schedule.

In the case of a server failure or other form of data loss, the DBA will use existing backups to restore lost information to the system. Different types of failures may require different recovery strategies, and the DBA must be prepared for any eventuality. With technology change, it is becoming ever more typical for a DBA to backup databases to the cloud, Oracle Cloud for Oracle Databases and MS Azure for SQL Server.

5. Security

A DBA needs to know potential weaknesses of the database software and the company's overall system and work to minimise risks. No system is one hundred per cent immune to attacks, but implementing best practices can minimise risks.

In the case of a security breach or irregularity, the DBA can consult audit logs to see who has done what to the data. Audit trails are also important when working with regulated data.

6. Authentication

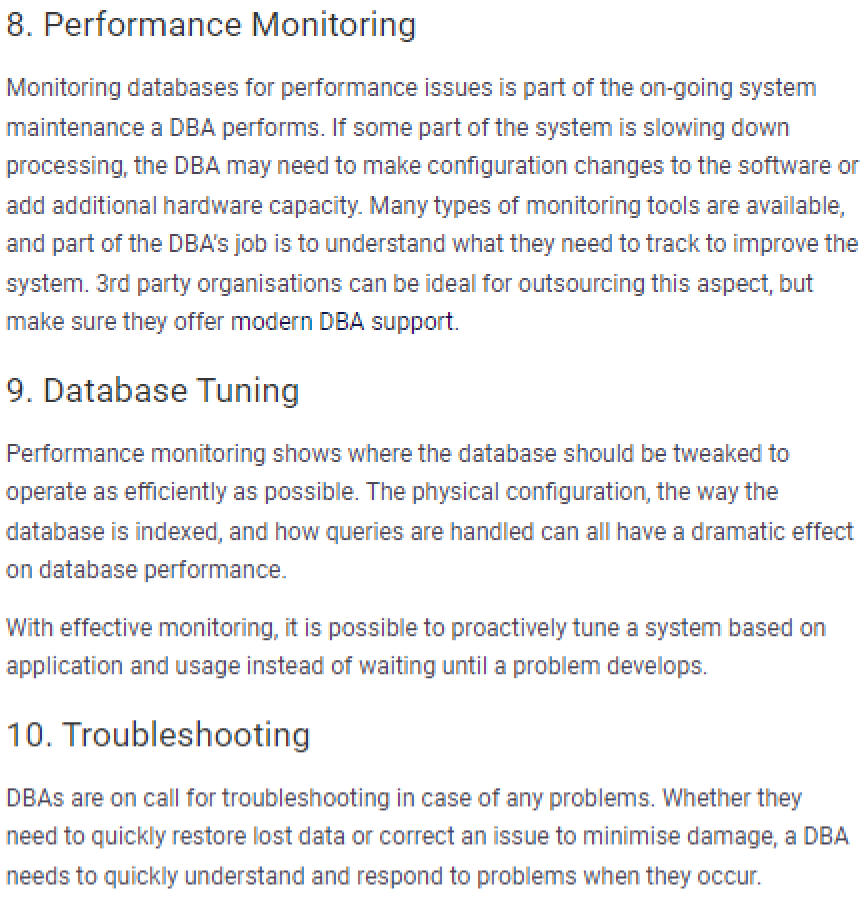
Setting up employee access is an important aspect of database security. DBAs control who has access and what type of access they are allowed. For instance, a user may have permission to see only certain pieces of information, or they may be denied the ability to make changes to the system.

7. Capacity Planning

The DBA needs to know how large the database currently is and how fast it is growing in order to make predictions about future needs. Storage refers to how much room the database takes up in server and backup space. Capacity refers to usage level.

If the company is growing quickly and adding many new users, the DBA will have to create the capacity to handle the extra workload.

Solve 4 Questions out of 6



8. Performance Monitoring

Monitoring databases for performance issues is part of the on-going system maintenance a DBA performs. If some part of the system is slowing down processing, the DBA may need to make configuration changes to the software or add additional hardware capacity. Many types of monitoring tools are available, and part of the DBA's job is to understand what they need to track to improve the system. 3rd party organisations can be ideal for outsourcing this aspect, but make sure they offer modern DBA support.

9. Database Tuning

Performance monitoring shows where the database should be tweaked to operate as efficiently as possible. The physical configuration, the way the database is indexed, and how queries are handled can all have a dramatic effect on database performance.

With effective monitoring, it is possible to proactively tune a system based on application and usage instead of waiting until a problem develops.

10. Troubleshooting

DBAs are on call for troubleshooting in case of any problems. Whether they need to quickly restore lost data or correct an issue to minimise damage, a DBA needs to quickly understand and respond to problems when they occur.

Solve 4 Questions out of 6

9. Database Schema

What is a Schema in SQL Server?

A Schema in SQL is a collection of database objects associated with a database. The username of a database is called a Schema owner (owner of logically grouped structures of data). Schema always belong to a single database whereas a database can have single or multiple schemas. Also, it is also very similar to separate namespaces or containers, which stores database objects. It includes various database objects including your tables, views, procedures, index, etc.

Let's move ahead and look at some of the advantages of using Schema in SQL.

Advantages of using Schema

- You can apply security permissions for separating and protecting database objects based on user access rights.
- A logical group of database objects can be managed within a database. Schemas play an important role in allowing the database objects to be organized into these logical groups.
- The schema also helps in situations where the database object name is the same. But these objects fall under different logical groups.
- A single schema can be used in multiple databases.
- The schema also helps in adding security.
- It helps in manipulating and accessing the objects which otherwise is a complex method.
- You can also transfer the ownership of several schemas.
- The objects created in the database can be moved among schemas.

These were few advantages, now the next topic is the method to create a schema.

How to create a Schema?

Syntax to create SQL:

```
CREATE SCHEMA [schema_name] [AUTHORIZATION owner_name]
[DEFAULT CHARACTER SET char_set_name]
[PATH schema_name[, ...]]
[ ANSI CREATE statements [...] ]
[ ANSI GRANT statements [...] ];
```

Solve 4 Questions out of 6

10. View, types of views/Index, types of indexes

What is a View? SQL has a special version of tables called View, which is a virtual table that is compiled in runtime. A View is just an SQL statement, and the data associated with it is not physically stored in the view but is stored in the base tables of it. It can contain all the rows and columns of a table or only a few selected rows and columns if there is a need to restrict the access. Depending on the written SQL query used to create the view, it can be created from one or many tables. Views can be used to structure data in ways for users to find it natural, simplify complex queries, restrict access to data, and summarize data from several tables to create reports.

Types of Views: There are two types of views in the SQL Server, namely System Defined Views and User Defined Views.

1. System Defined Views: The System Defined Views are predefined views that already exist in the SQL Server database, such as Tempdb, Master, and temp. Each of the databases has its own properties and functions. The template database for all User Defined views is from the Master database. It contains many predefined views that are templates for tables and other databases. It contains nearly 230 of the predefined views. System Defined Views will be automatically attached to all User Defined databases. And these provide information about the database, tables, and all the properties of the database and tables. There are three types of System defined views, Information Schema, Catalog View, and Dynamic Management View.

❑ Information Schema There are twenty different schema views in the SQL server. They are used to display the physical information of the database, such as tables, constraints, columns, and views. This view starts with INFORMATION_SCHEMA and followed by the View Name.

❑ Catalog View These are used to return information used by the SQL server. Catalog views provide an efficient way to obtain, present, and transform custom forms of information. But they do not include any information about backup, replication, or maintenance plans, etc. These views are used to access metadata of databases, and the names and column names are descriptive, helping a user to query what is expected.

Solve 4 Questions out of 6

❑ Dynamic Management View These were introduced in the SQL server in 2005. The administer can get information about the server state to diagnose problems, monitor the health of the server instance, and tune performance through these views. The Server-scoped Dynamic Management View is only stored in the Master database, whereas the Database-scoped Dynamic Management View is stored in each database.

2. User Defined Views: These are the types of views that are defined by the users. There are two types under User Defined views, Simple View and Complex View.

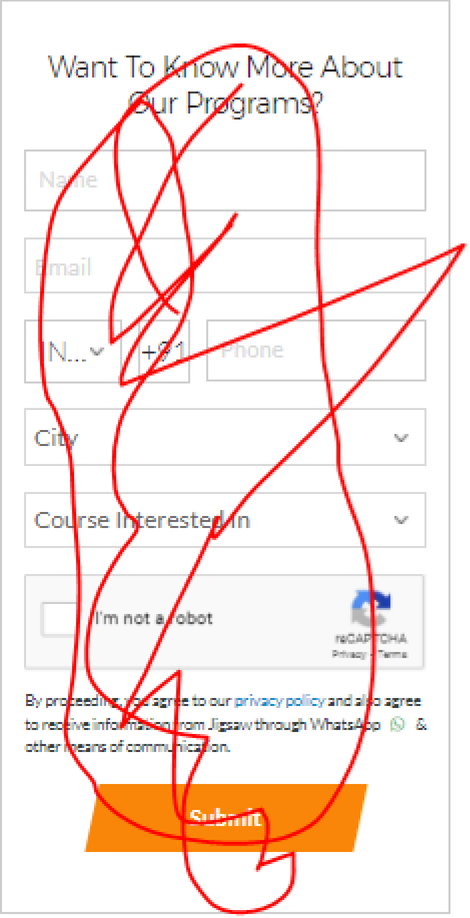
❑ Simple View These views can only contain a single base table or can be created only from one table. Group functions such as MAX (), COUNT (), etc., cannot be used here, and it does not contain groups of data. By using Simple View, DML operations can be performed. Insert, delete, and update are directly possible, but Simple View does not contain group by, pseudocolumn like rownum, distinct, columns defined by expressions. Simple view also does not include NOT NULL columns from the base tables.

❑ Complex View These views can contain more than one base table or can be constructed on more than one base table, and they contain a group by clause, join conditions, an order by clause. Group functions can be used here, and it contains groups of data. Complex views cannot always be used to perform DML operations. Insert, delete, and update cannot be applied directly on complex views.

Solve 4 Questions out of 6

INTRODUCTION

The index is named as a design in SQL server stored or maintained wilt in-memory structure or on disk related with a table or views, which is utilized principally to recognize a specific set or a row Table or Views. Indexes in SQL are the individual lookup tables, which are utilized by the data set internet searcher to accelerate the general information recovery.



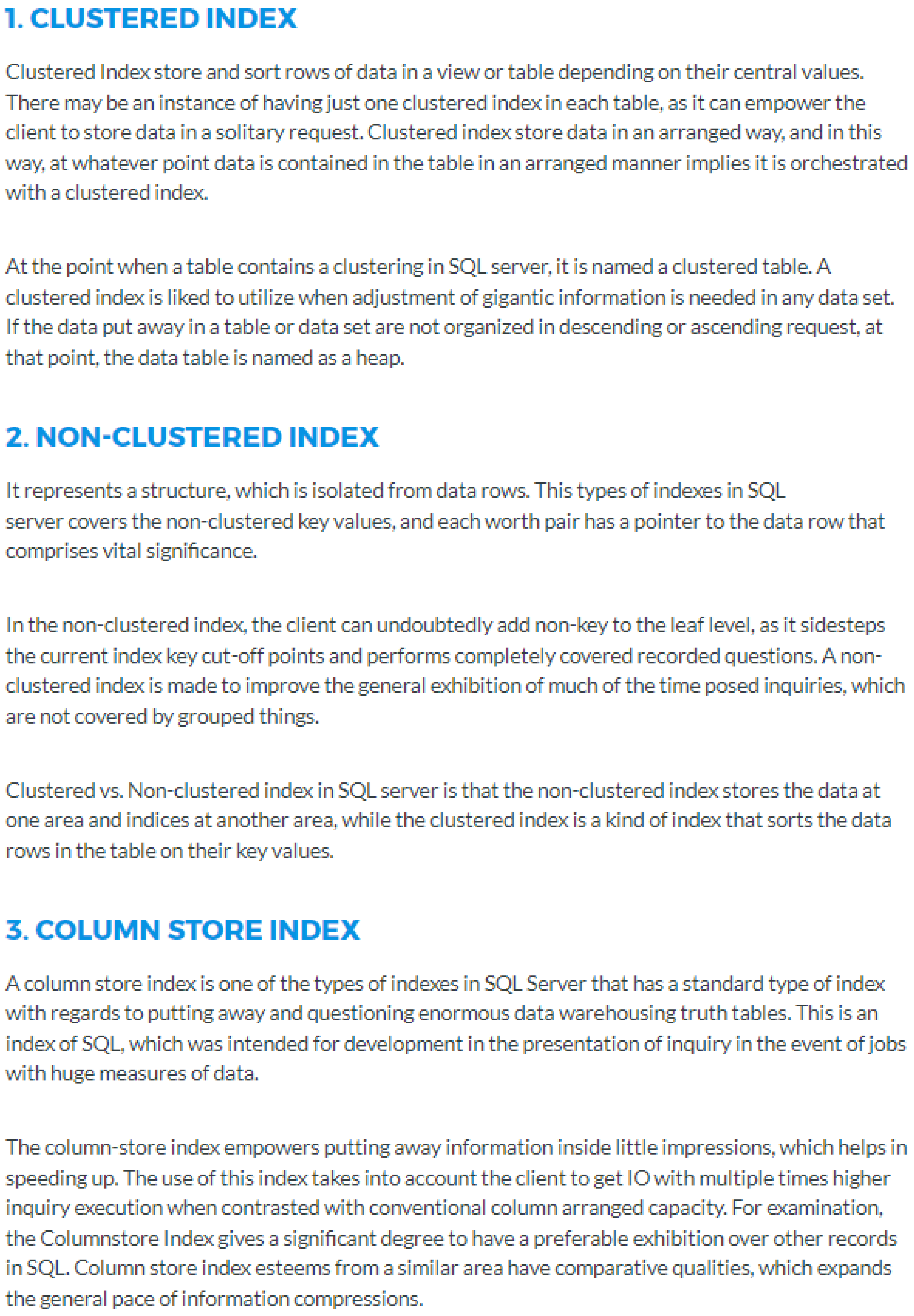
The use of the index in SQL is to rapidly discover the data in a data set table without looking through each row of it. In SQL Index, it is basic to keep up more extra storage to make a copy duplicate of the data set. Tables in SQL server are contained inside database item holders that are called Schemas. The schema likewise fills in as a security limit, where you can restrict data set client authorizations to be on a particular schema level as it were. To know what are the different types of Indexes in SQL Server, then read this article to explore them and have a better understanding of them.

DIFFERENT TYPES OF INDEXES IN SQL SERVER

There are various types of indexes in SQL server:

1. Clustered Index
2. Non-Clustered Index
3. Column Store Index
4. Filtered Index
5. Hash Index
6. Unique Index

Solve 4 Questions out of 6



1. CLUSTERED INDEX

Clustered Index store and sort rows of data in a view or table depending on their central values. There may be an instance of having just one clustered index in each table, as it can empower the client to store data in a solitary request. Clustered index store data in an arranged way, and in this way, at whatever point data is contained in the table in an arranged manner implies it is orchestrated with a clustered index.

At the point when a table contains a clustering in SQL server, it is named a clustered table. A clustered index is liked to utilize when adjustment of gigantic information is needed in any data set. If the data put away in a table or data set are not organized in descending or ascending request, at that point, the data table is named as a heap.

2. NON-CLUSTERED INDEX

It represents a structure, which is isolated from data rows. This types of indexes in SQL server covers the non-clustered key values, and each worth pair has a pointer to the data row that comprises vital significance.

In the non-clustered index, the client can undoubtedly add non-key to the leaf level, as it sidesteps the current index key cut-off points and performs completely covered recorded questions. A non-clustered index is made to improve the general exhibition of much of the time posed inquiries, which are not covered by grouped things.

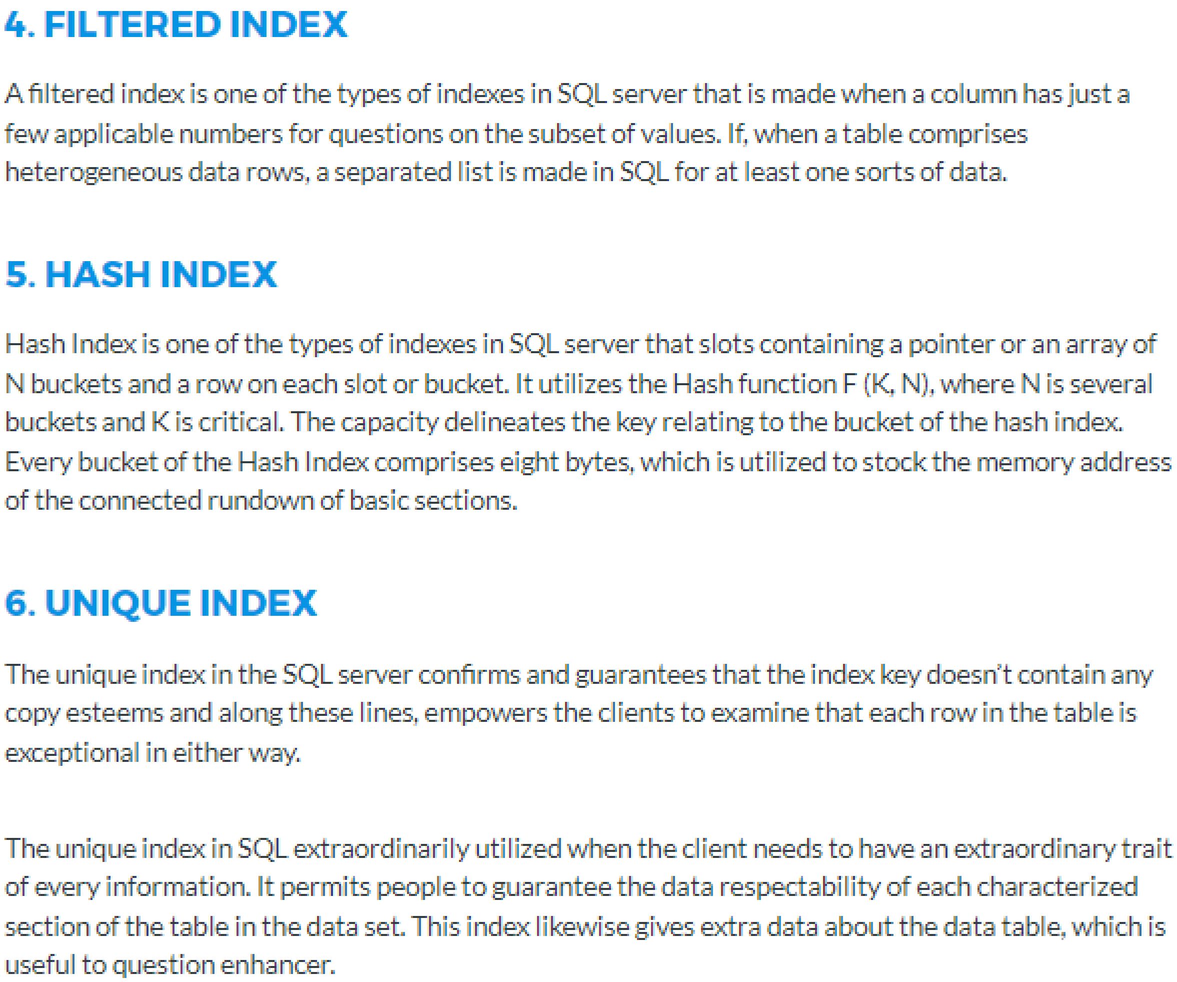
Clustered vs. Non-clustered index in SQL server is that the non-clustered index stores the data at one area and indices at another area, while the clustered index is a kind of index that sorts the data rows in the table on their key values.

3. COLUMN STORE INDEX

A column store index is one of the types of indexes in SQL Server that has a standard type of index with regards to putting away and questioning enormous data warehousing truth tables. This is an index of SQL, which was intended for development in the presentation of inquiry in the event of jobs with huge measures of data.

The column-store index empowers putting away information inside little impressions, which helps in speeding up. The use of this index takes into account the client to get IO with multiple times higher inquiry execution when contrasted with conventional column arranged capacity. For examination, the Columnstore Index gives a significant degree to have a preferable exhibition over other records in SQL. Column store index esteems from a similar area have comparative qualities, which expands the general pace of information compressions.

Solve 4 Questions out of 6



4. FILTERED INDEX

A filtered index is one of the types of indexes in SQL server that is made when a column has just a few applicable numbers for questions on the subset of values. If, when a table comprises heterogeneous data rows, a separated list is made in SQL for at least one sorts of data.

5. HASH INDEX

Hash Index is one of the types of indexes in SQL server that slots containing a pointer or an array of N buckets and a row on each slot or bucket. It utilizes the Hash function F (K, N), where N is several buckets and K is critical. The capacity delineates the key relating to the bucket of the hash index. Every bucket of the Hash Index comprises eight bytes, which is utilized to stock the memory address of the connected rundown of basic sections.

6. UNIQUE INDEX

The unique index in the SQL server confirms and guarantees that the index key doesn't contain any copy esteems and along these lines, empowers the clients to examine that each row in the table is exceptional in either way.

The unique index in SQL extraordinarily utilized when the client needs to have an extraordinary trait of every information. It permits people to guarantee the data respectability of each characterized section of the table in the data set. This index likewise gives extra data about the data table, which is useful to question enhancer.

THE END!