# Indexing in Databases:

## What is Indexing?

**Indexing** is a data structure technique which allows you to quickly retrieve records from a database file. An Index is a small table having only two columns. The first column comprises a copy of the primary or candidate key of a table. Its second column contains a set of [pointers](https://www.guru99.com/c-pointers.html) for holding the address of the disk block where that specific key value stored.

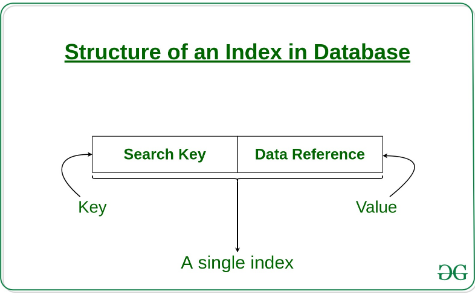
An index –

* Takes a search key as input
* Efficiently returns a collection of matching records.

# It is a data structure technique which is used to quickly locate and access the data in a database

Indexes are created using a few database columns.

* The first column is the **Search key** that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly.   
  *Note: The data may or may not be stored in sorted order.*
* The second column is the **Data Reference** or **Pointer** which contains a set of pointers ho

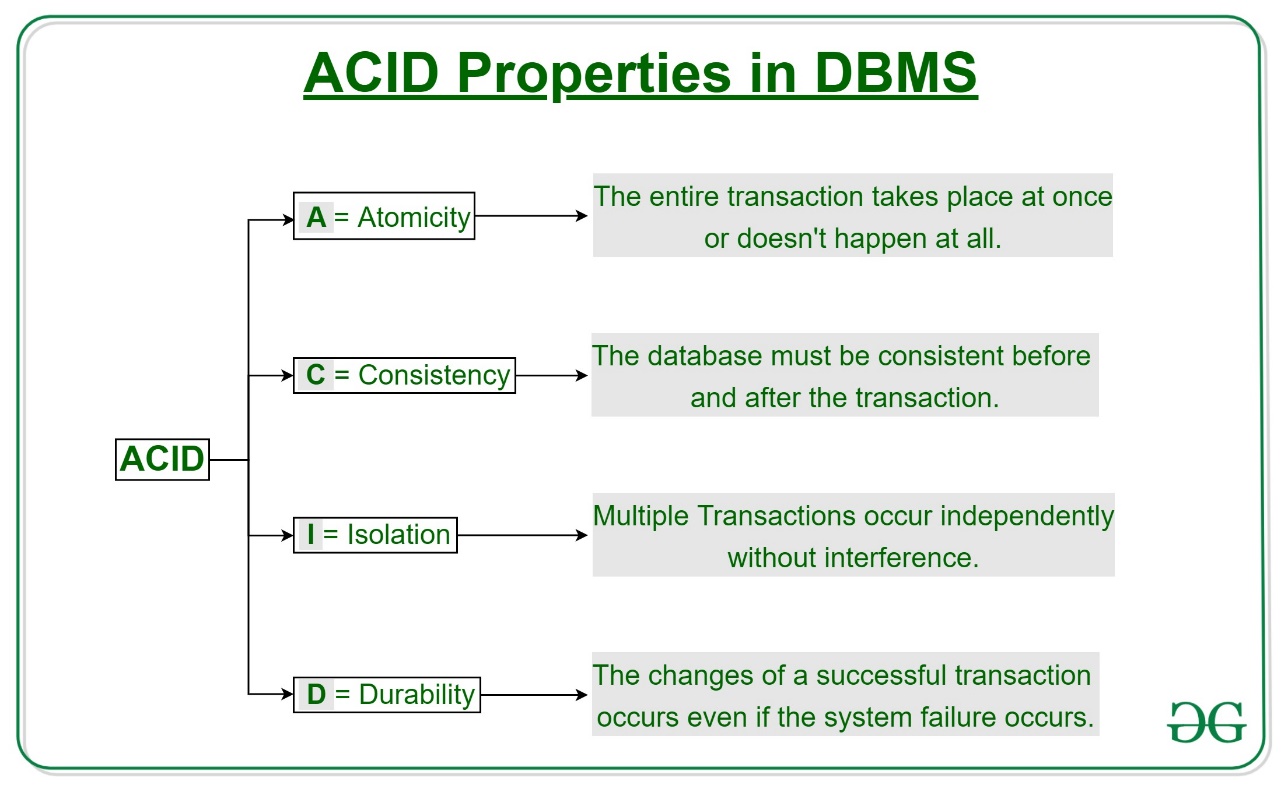


The indexing has various attributes:

* **Access Types**: This refers to the type of access such as value based search, range access, etc.
* **Access Time**: It refers to the time needed to find particular data element or set of elements.
* **Insertion Time**: It refers to the time taken to find the appropriate space and insert a new data.
* **Deletion Time**: Time taken to find an item and delete it as well as update the index structure.
* **Space Overhead**: It refers to the additional space required by the index

# ACID Properties in DBMS

ACID Properties in SQL Server ensure Data Integrity during a transaction. The ACID is an acronym for Atomicity, Consistency, Isolation, and Durability.



**Some important points:**

| **Property** | **Responsibility for maintaining properties** |
| --- | --- |
| Atomicity | Transaction Manager |
| Consistency | Application programmer |
| Isolation | Concurrency Control Manager |
| Durability | Recovery Manager |

**1. Relational Database :**   
RDBMS stands for Relational Database Management Systems. It is most popular database. In it, data is store in the form of row that is in the form of tuple. It contain numbers of table and data can be easily accessed because data is store in the table. This Model was proposed by E.F. Codd.

**2. NoSQL :**   
NoSQL Database stands for a non-SQL database. NoSQL database doesn’t use table to store the data like relational database. It is used for storing and fetching the data in database and generally used to store the large amount of data. It supports query language and provides better performance.

**Difference between Relational database and NoSQL :**

|  |  |
| --- | --- |
| Relational Database | NoSQL |
| It is used to handle data coming in low velocity. | It is used to handle data coming in high velocity. |
| It gives only read scalability. | It gives both read and write scalability. |
| It manages structured data. | It manages all type of data. |
| Data arrives from one or few locations. | Data arrives from many locations. |
| It supports complex transactions. | It supports simple transactions. |
| It has single point of failure. | No single point of failure. |
| It handles data in less volume. | It handles data in high volume. |
| Transactions written in one location. | Transactions written in many locations. |
| support ACID properties compliance | doesn’t support ACID properties |
| Its difficult to make changes in database once it is defined | Enables easy and frequent changes to database |
| schema  is mandatory to store the data | schema design is not required |
| Deployed in vertical fashion. | Deployed in Horizontal fashion. |

### Features of relational databases

* They work with structured data.
* Relationships in the system have constraints, which promotes a high level of data integrity.
* There are limitless indexing capabilities, which results in faster query response times.
* They are excellent at keeping data transactions secure.
* They provide the ability to write complex SQL queries for data analysis and reporting.
* Their models can ensure and enforce business rules at the data layer adding a level of data integrity not found in a non-relational database.
* They are table and row oriented.
* They Use SQL (structured query language) for shaping and manipulating data, which is very powerful.
* SQL database examples: [MySql](https://www.pluralsight.com/paths/mysql), [Oracle](https://www.pluralsight.com/courses/oracle-database-12c-fundamentals), Sqlite, Postgres and MS-SQL. NoSQL database examples: MongoDB, [BigTable](https://www.pluralsight.com/courses/google-bigtable-architecting-big-data-solutions), Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb.
* SQL databases are best fit for heavy duty transactional type applications.

### Features of non-relational databases

* They have the ability to store large amounts of data with little structure.
* They provide scalability and flexibility to meet changing business requirements.
* They provide schema-free or schema-on-read options.
* They have the ability to capture all types of data “Big Data” including unstructured data.
* They are document oriented.
* NoSQL or non-relational databases examples: MongoDB, Apache Cassandra, Redis, Couchbase and Apache HBase.
* They are best for Rapid Application Development. NoSQL is the best selection for flexible data storage with little to no structure limitations.
* They provide flexible data model with the ability to easily store and combine data of any structure without the need to modify a schema.

### 1) In what sequence SQL statements are processed?

The clauses of the select are processed in the following sequence

1. FROM clause
2. WHERE clause
3. GROUP BY clause
4. HAVING clause
5. SELECT clause
6. ORDER BY clause
7. TOP clause

### 2) Can we write a distributed query and get some data that is located on another server and on Oracle Database?

[SQL Server](https://mindmajix.com/sql-server-tutorial) can be lined to any server provided it has an OLE-DB provider from Microsoft to allow a link.

For E.g. Oracle has an OLE-DB provider for oracle that Microsoft provides to add it as a linked server to the SQL Server group.

### 3) If we drop a table, does it also drop related objects like constraints, indexes, columns, defaults, Views, and Stored Procedures?

**YES**, SQL Server drops all related objects, which exist inside a table like constraints, indexes, columns, defaults, etc. BUT dropping a table will not drop Views and Stored Procedures as they exist outside the table.

How would you determine the time zone under which a database was operating?

### 4) Can we add an identity column to the decimal datatype?

**YES**, SQL Server support this

### 5) What is the difference between LEFT JOIN with WHERE clause & LEFT JOIN with nowhere clause?

OUTER LEFT/RIGHT JOIN with WHERE clause can act like an INNER JOIN if not used wisely or logically.

[](https://bit.ly/3if9dmk)

### 6) What are the multiple ways to execute a dynamic query?

* EXEC sp\_executesql,
* EXECUTE()

### 7) What is the Difference between COALESCE() & ISNULL()?

ISNULL accepts only 2 parameters. The first parameter is checked for a NULL value, if it is NULL then the second parameter is returned, otherwise, it returns the first parameter.  
  
COALESCE accepts two or more parameters. One can apply 2 or as many parameters, but it returns only the first non NULL parameter,

### 8) How do you generate file output from SQL?

While using SQL Server Management Studio or Query Analyzer, we have an option in Menu BAR.QUERTY >> RESULT TO >> Result to FILE

### 9) How do you prevent SQL Server from giving you informational messages during and after a SQL statement execution?

SET NOCOUNT OFF

### 10) By Mistake, Duplicate records exists in a table, how can we delete the copy of a record?

with T as

(

select \* , row\_number() over (partition by Emp\_ID order by Emp\_ID) as rank

from employee

)

delete

from T

where rank > 1

### 11) WHAT OPERATOR PERFORMS PATTERN MATCHING?

The pattern matching operator is LIKE and it has to use with two attributes

1. %  means matches zero or more characters and

2. \_ ( underscore ) means matching exactly one character

### 12) What’s the logical difference, if any, between the following SQL expressions?

-- Statement 1

SELECT COUNT ( \* ) FROM Employees

-- Statement 2

SELECT SUM ( 1 ) FROM Employees

They’re the same unless table Employee table is empty, in which case the first yields a one-column, a one-row table containing zero, and the second yields a one-column, one-row table "containing a null."

### 13) Is it possible to update the Views? If yes, How, If Not, Why?

Yes, We can modify views but a DML statement on a join view can modify only one base table of the view (so even if the view is created upon a join of many tables, only one table, the key preserved table can be modified through the view).

### 14) Could you please name different kinds of Joins available in SQL Server?

* OUTER JOIN – LEFT, RIGHT, CROSS, FULL ;
* INNER JOIN

### 15) How important do you consider cursors or while loops for a transactional database?

would like to avoid cursor in the OLTP database as much as possible, Cursors are mainly only used for maintenance or warehouse operations.

### 16) What is a correlated subquery?

When a subquery is tied to the outer query. Mostly used in self joins.

### 17) What is faster, a correlated subquery or an inner join?

Correlated subquery.

### 18) You are supposed to work on SQL optimization and given a choice which one runs faster, a correlated subquery or an exists?

Exists

### 19) Can we call. DLL from the SQL server?

YES, We can call. Dll from SQL Server.

### 20) What are the pros and cons of putting a scalar function in a queries select list or in the where clause?

Should be avoided if possible as Scalar functions in these places make the query slow down dramatically.

### 21) What are user-defined data types and when you should go for them?

User-defined data types let you extend the base SQL Server data types by providing a descriptive name, and format to the database. Take for example, in your database, there is a column called Flight\_Num which appears in many tables. In all these tables it should be varchar(8). In this case, you could create a user-defined data type called Flight\_num\_type of varchar(8) and use it across all your tables.

See sp\_addtype, sp\_droptype in books online.

### 22) Can You Explain Integration Between SQL Server 2005 And Visual Studio 2005?

This integration provides a wider range of development with the help of CLR for database servers because CLR helps developers to get flexibility for developing database applications and also provides language interoperability just like Visual C++, Visual Basic .Net and Visual C# .Net.

The CLR helps developers to get the arrays, classes and exception handling available through programming languages such as Visual C++ or Visual C# which is used in stored procedures, functions and triggers for creating database application dynamically and also provide more efficient reuse of code and faster execution of complex tasks. We particularly liked the error-checking powers of the CLR environment, which reduces run-time errors

### 23) What are Index, cluster index, and non-cluster index?

**Clustered Index:-** A Clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore table may have only one clustered index.

**Non-NonClustered Index:-**A Non-Clustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows in the disk. The leaf nodes of a non-clustered index do not consist of the data pages. instead, the leaf node contains index rows.

### 24) Write down the general syntax for a SELECT statement covering all the options.

Here’s the basic syntax: (Also checkout SELECT in books online for advanced syntax)

SELECT select\_list

[INTO new\_table\_]

FROM table\_source

[WHERE search\_condition]

[GROUP BY group\_by\_\_expression]

[HAVING search\_condition]

[ORDER BY order\_\_expression [ASC | DESC] ]

### 25). What is a join and explain different types of joins?

Joins are used in queries to explain how different tables are related. Joins also let you select data from a table depending upon data from another table.

Types of joins:

INNER JOINs,

OUTER JOINs,

CROSS JOINs

OUTER JOINs are further classified as

LEFT OUTER JOINS,

RIGHT OUTER JOINS and

FULL OUTER JOINS.

For more information see pages from books online titled: "Join Fundamentals" and "Using Joins".

### 26) What is the OSQL utility?

OSQL is a command-line tool that is used to execute the query and display the result the same as a query analyzer but everything is in the command prompt.

### 27) What Is the Difference Between OSQL And Query Analyzer?

OSQL is the command-line tool that executes the query and displays the result the same as a query analyzer but the query analyzer is graphical and OSQL is a command-line tool. OSQL is quite useful for batch processing or executing remote queries.

### 28) What Is Cascade delete/update?

CASCADE allows deletions or updates of key values to cascade through the tables defined to have foreign key relationships that can be traced back to the table on which the modification is performed.

## SQL Server Interview Questions For 2-5 Years Experienced

### 29) What are some of the join algorithms used when SQL Server joins tables.

1. Loop Join (indexed keys unordered)
2. Merge Join (indexed keys ordered)
3. Hash Join (non-indexed keys)

### 30) What is the maximum number of tables that can join in a single query?

256, check SQL Server Limits

### 31) What are Magic Tables in SQL Server?

The MAGIC tables are automatically created and dropped, in case you use TRIGGERS. SQL Server has two magic tables named, INSERTED and DELETED

These are maintained by the SQL server for their Internal processing. When we use update insert or delete on tables these magic tables are used. These are not physical tables but are Internal tables. Whenever we use insert statement is fired the Inserted table is populated with newly inserted Row and whenever delete statement is fired the Deleted table is populated with the deleted row.

But in case of update statement is fired both Inserted and Deleted tables used for records the Original row before updating get stored in the Deleted table and the new row Updated gets store in Inserted table.

### 32) Can we disable a trigger? if yes HOW?

YES, we can disable a single trigger on the database by using  “DISABLE TRIGGER triggerName ON <>”

we also have an option to disable all the triggers by using, “DISABLE Trigger ALL ON ALL SERVER”

### 33) Why do you need indexing? where is Stored and what do you mean by schema object? For what purpose we are using view?

We can’t create an Index on an Index... The index is stored in the user\_index table. Every object that has been created on Schema is Schema Object like Table, View etc. If we want to share the particular data to various users we have to use the virtual table for the Base table. So that is a view.

Indexing is used for faster search or to retrieve data faster from the various tables. Schema containing a set of tables, basically schema means logical separation of the database. The view is crated for faster retrieval of data. It’s a customized virtual table. we can create a single view of multiple tables. Only the drawback is..view needs to be get refreshed for retrieving updated data.

### 34) What is the difference between UNION and UNION ALL?

Union will remove the duplicate rows from the result set while Union all doesn't.

### 35) Which system table contains information on constraints on all the tables created?

USER\_CONSTRAINTS,

system table contains information on constraints on all the tables created

## SQL Server Joins Interview Questions

### 35) What are the different Types of Join?

Below are the different types of [SQL Server Joins](https://mindmajix.com/joins-sql-server):

1. **Cross Join:** A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. A common example is when a company wants to combine each product with a pricing table to analyze each product at each price.
2. **Inner Join:** A join that displays only the rows that have a match in both joined tables is known as inner Join. This is the default type of join in the Query and View Designer.
3. **Outer Join:** A join that includes rows even if they do not have related rows in the joined table is an Outer Join. You can create three different outer joins to specify the unmatched rows to be included:
   * **Left Outer Join:** In Left Outer Join all rows in the first-named table i.e. "left" table, which appears leftmost in the JOIN clause are included. Unmatched rows in the right table do not appear.
   * **Right Outer Join:** In Right Outer Join all rows in the second-named table i.e. "right" table, which appears rightmost in the JOIN clause are included. Unmatched rows in the left table are not included.
   * **Full Outer Join:** In Full Outer Join all rows in all joined tables are included, whether they are matched or not.
4. **Self Join:** This is a particular case when one table joins to itself, with one or two aliases to avoid confusion. A self-join can be of any type, as long as the joined tables are the same. A self-join is rather unique in that it involves a relationship with only one table. A common example is when a company has a hierarchal reporting structure whereby one member of staff reports to another. Self Join can be Outer Join or Inner Join.

### 36) What is Data-Warehousing?

1. **Subject-oriented**, meaning that the data in the database is organized so that all the data elements relating to the same real-world event or object are linked together;
2. **Time-variant**, meaning that the changes to the data in the database are tracked and recorded so that reports can be produced showing changes over time;
3. **Non-volatile**, meaning that data in the database is never over-written or deleted, once committed, the data is static, read-only, but retained for future reporting.
4. **Integrated**, meaning that the database contains data from most or all of an organization’s operational applications and that this data is made consistent.

### 37) What is a live lock?

A livelock is one, where a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keeps interfering. SQL Server detects the situation after four denials and refuses further shared locks. A livelock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely.

### 38) How SQL Server executes a statement with nested subqueries?

When SQL Server executes a statement with nested subqueries, it always executes the innermost query first. This query passes its results to the next query and so on until it reaches the outermost query. It is the outermost query that returns a result set.

### 39) How do you add a column to an existing table?

ALTER TABLE Department ADD (AGE, NUMBER);

### 40) Can one drop a column from a table?

**YES**, to delete a column in a table, use  ALTER TABLE table\_name DROP COLUMN column\_name

### 41) Which statement do you use to eliminate padded spaces between the month and day values in a function TO\_CHAR(SYSDATE,’Month, DD, YYYY’)?

To remove padded spaces, you use the "fm" prefix before the date element that contains the spaces. TO\_CHAR(SYSDATE,’fmMonth DD, YYYY’)

### 42) Which operator do you use to return all of the rows from one query except rows are returned in a second query?

You use the **EXCEPT**operator to return all rows from one query except where duplicate rows are found in a second query. The UNION operator returns all rows from both queries minus duplicates. The UNION ALL operator returns all rows from both queries including duplicates. The INTERSECT operator returns only those rows that exist in both queries.

### 43) How will you create a column alias?

The **AS**keyword is optional when specifying a column alias.

### 44) In what sequence SQL statements are processed?

The clauses of the subselect are processed in the following sequence (DB2):

1. FROM clause
2. WHERE clause
3. GROUP BY clause
4. HAVING clause
5. SELECT clause
6. ORDER BY clause
7. FETCH FIRST clause

### 45) How can we determine what objects a user-defined function depends upon?

**sp\_depends**system stored procedure or query the says depends on system table to return a list of objects that a user-defined function depends upon

SELECT DISTINCT so1.name, so2.name FROM sysobjects so1

INNER JOIN sysdepends sd

ON so1.id = sd.id

INNER JOIN sysobjects so2

ON so2.id = sd.depid

WHERE so1.name = '<>'

### 46). What is lock escalation?

A query first takes the lowest level lock possible with the smallest footprint (row-level). When too many rows are locked (requiring too much RAM) the lock is escalated to a range or page lock. If too many pages are locked, it may escalate to a table lock.

### 47) What are the main differences between #temp tables and @table variables and which one is preferred?

1. SQL Server can create column statistics on #temp tables
2. Indexes can be created on #temp tables
3. @table variables are stored in memory up to a certain threshold.

### 48) What are Checkpoint In SQL Server?

When we did the operation on SQL SERVER that is not committed directly to the database. All operations must be logged in to Transaction Log files after that they should be done on to the main database.CheckPoint is the point that alerts SQL Server to save all the data to the main database if no checkpoint is there then log files get full we can use the Checkpoint command to commit all data in the SQL SERVER. When we stop the SQL Server it will take a long time because Checkpoint is also fired.

|  |
| --- |
| ***Read these latest***[***SQL Interview Questions For 5+ Years Experienced***](https://mindmajix.com/sql-server-interview-questions-for-experienced)***that helps you grab high-paying jobs*** |

### 49) Why we use the OPEN XML clause?

OPENXML parses the XML data in SQL Server in an efficient manner. Its primary ability is to insert XML data into the DB.

### 50) Can we store PDF files inside the SQL Server table?

**YES,**we can store this sort of data using a blob datatype.

### 51) Can we store Videos inside the SQL Server table?

**YES,**we can store Videos inside SQL Server by using FILESTREAM data type, which was introduced in SQL Server 2008.

### 52) Can we hide the definition of a stored procedure from a user?

YES, while creating stored procedure we can use WITH ENCRYPTION which will convert the original text of the CREATE PROCEDURE statement to an encrypted format.

### 53) What have included columns when we talk about SQL Server indexing?

Indexed with included columns were developed in SQL Server 2005 that assists in covering queries. Indexes with Included Columns are non clustered indexes that  
  
have the following benefits:

* Columns defined in the include statement, called non-key columns, are not counted in the  
  a number of columns by the Database Engine.
* Columns that previously could not be used in queries, like nvarchar(max), can be included  
  as a non-key column.
* A maximum of 1023 additional columns can be used as non-key columns.

### 54). What is an execution plan? How would you view the execution plan?

An execution plan is basically a road map that graphically or textually shows the data retrieval methods chosen by the SQL Server query optimizer for a stored procedure or ad-hoc query and is a very useful tool for a developer to understand the performance characteristics of a query or stored procedure since the plan is the one that SQL Server will place in its cache and use to execute the stored procedure or query.

From within Query Analyzer is an option called "Show Execution Plan" (located on the Query drop-down menu). If this option is turned on it will display the query execution plan in a separate window when the query is run again.

### 55). Explain UNION, MINUS, UNION ALL, INTERSECT?

* **INTERSECT**: returns all distinct rows selected by both queries.
* **MINUS:**returns all distinct rows selected by the first query but not by the second.
* **UNION:** returns all distinct rows selected by either query
* **UNION  ALL:**  returns  all  rows  selected  by  either query, including all duplicates

|  |
| --- |
| → Explore [***SQL Server Sample Resumes***](https://mindmajix.com/sql-server-sample-resumes) Download & Edit, Get Noticed by Top Employers! |

## SQL Server Query Interview Questions with Answers

**SQL Server DATEADD() Function**

### 56) Write a Query to display the date after 15 days?

SELECT DATEADD(dd, 15,getdate())

### 57) Write a Query to display the date after 12 months?

SELECT DATEADD(mm, 2, getdate())

### 58) Write a Query to display the date before 15 days?

SELECT DATEADD(dd, -15, getdate())

**SQL Server DATEDIFF() Function**

### 59) Write a Query to display employee details along with exp?

SELECT \*

DATEDIFF(yy, doj, getdate()) AS ‘Exp’ FROM employee

### 60) Write a Query to display employee details who is working in ECE department & who his having more than 3 years of exp?

SELECT \*

DATEDIFF(yy, doj, getdate()) AS ‘Exp’

FROM employee

WHERE DATEDIFF(yy, doj, getdate())>3 AND dept\_name=’ECE’

### 61) Write a Query to display employee details along with age?

SELECT \*

DATEDIFF(yy, dob, getdate()) AS ‘Age’ FROM employee

### 62) Write a Query to display employee details whose age >18?

SELECT \*

DATEDIFF(yy, dob, getdate()) AS ‘Age’ FROM employee

WHERE DATEDIFF(yy, dob, getdate())>18

**SQL Server Multi-Row Functions**

### 63) Write a Query to display the minimum salary of an employee?

SELECT MIN (salary)

FROM employee

### 64) Write a Query to display the maximum salary of an employee?

SELECT MAX(salary)

FROM employee

### 65) Write a Query to display the total salary of all employees?

SELECT SUM(salary) FROM employee

### 66) Write a Query to display the average salary of an employee?

SELECT AVG(salary) FROM employee

### 67) Write a Query to count the number of employees working in the company?

SELECT COUNT(\*) FROM employee

### 68) Write a Query to display the minimum & maximum salary of the employee?

SELECT MIN(salary) AS ‘min sal’, MAX(salary) AS ‘max sal’ FROM employee

### 69) Write a Query to count the number of employees working in the ECE department?

SELECT COUNT(\*) FROM employee WHERE dept\_name=’ECE’

### 70) Write a Query to display the second max salary of an employee?

SELECT MAX(salary)

FROM employee

WHERE salary < (SELECT MAX(salary) FROM emp)

### 71) Write a Query to display the third max salary of an employee?

SELECT MAX(salary)

FROM employee

WHERE salary < (SELECT MAX(salary) FROM emp where salary < (SELECT MAX(salary) FROM emp))

**SQL SERVER: GROUP BY Clause**

### 72) Write a Query to display the total salary of employees based on the city?

SELECT city, SUM(salary)

FROM employee

GROUP BY city;

### 73) Write a Query to display a number of employees based on the city?

SELECT city, COUNT(emp\_no)

FROM employee

GROUP BY city;

**(OR)**

SELECT city, COUNT(emp\_no) AS ‘no.of employees’

FROM employee

GROUP BY city;

### 74) Write a Query to display the total salary of employees based on region?

SELECT region, SUM(salary) AS ‘total\_salary’

FROM employee

GROUP BY region;

### 75) Write a Query to display the number of employees working in each region?

SELECT region, COUNT(gender)

FROM employee

GROUP BY region;

**(OR)**

SELECT region, COUNT(gender) AS ‘no.of males’

FROM employee

GROUP BY region;

### 76) Write a Query to display minimum salary & maximum salary based on dept\_name?

SELECT dept\_name, MIN(salary) AS ‘min sal’, MAX(salary) AS ‘max sal’

FROM employee

GROUP BY dept\_name

### 77) Write a Query to display the total salary of employees based on dept\_name?

SELECT dept\_name, SUM(salary) AS ‘total\_sal’

FROM employee

GROUP BY dept\_name

### 78) Write a Query to display no. of males in each department?

SELECT dept\_name, COUNT(gender)

FROM employee

GROUP BY dept\_name

WHERE gender=’male’

**(OR)**

SELECT dept\_name, COUNT(gender) AS ‘no.of males’

FROM employee

WHERE gender=’male’

GROUP BY dept\_name;

**Note:**We cannot apply where condition in GROUP BY CLAUSE if we want to apply use having clause.

We have to use WHERE condition before GROUP BY but cannot apply where condition after GROUP BY.

**SQL SERVER: Having Clause**

### 79) Write a Query to display the total salary of employees based on whose total salary > 12000?

SELECT city, SUM(salary) AS ‘total\_salary’

FROM employee

GROUP BY city

HAVING SUM(salary)>12000;

### 80) Write a Query to display the total salary of all employees based on a city whose average salary >= 23000?

SELECT city, SUM(salary) AS ‘total\_salary’

FROM employee

GROUP BY city

HAVING AVG(salary) >= 23000;

**SQL SERVER: SUB QUERIES**

### 81) Write a Query to display employee details whose employee numbers are 101, 102?

SELECT \*

FROM employee

WHERE Emp\_No in (101, 102)

**(OR)**

SELECT \* FROM employee

WHERE Emp\_No in (select emp\_no from emp)

### 82) Write a Query to display employee details belongs to the ECE department?

SELECT Emp\_No, Emp\_Name, Salary

FROM employee

WHERE dept\_no in (select dept\_no from dept where dept\_name = ‘ECE’)

**SQL SERVER TOP Clause**

### 83) Write a Query to display the first record from the table?

SELECT TOP 1 \*

FROM employee

### 84) Write a Query to display the top 3 records from the table?

SELECT TOP 3 \*

FROM employee

### 85) Write a Query to display the last record from the table?

SELECT TOP 1 \*

FROM employee

ORDER BY emp\_no descending

**SQL SERVER: Ranking Functions**

**Student Details Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Student\_No** | **Student\_Name** | **Percentage** | **Row\_ID** | **Rank\_ID** | **DenseRank\_ID** |
| 105 | James | 87 | 1 | 1 | 1 |
| 106 | John | 83 | 2 | 2 | 2 |
| 101 | Anil | 83 | 3 | 2 | 2 |
| 104 | Vijay | 83 | 4 | 2 | 2 |
| 108 | Rakesh | 76 | 5 | 5 | 3 |
| 102 | Sunil | 76 | 6 | 5 | 3 |
| 103 | Ajay | 76 | 7 | 5 | 3 |
| 107 | Ram | 75 | 8 | 8 | 4 |

### 86) Write a Query to display student details along with the row\_no order by student name?

SELECT \*, ROW\_NUMBER() OVER (ORDER BYstudent\_name) AS ‘Row\_ID’

FROM employee

### 87) Write a Query to display even records from the table?

SELECT \* FROM ( SELECT \*, ROW\_NUMBER() OVER (ORDER BY student\_no) AS ‘ Row\_ID’ FROM student)

WHERE row\_id %2=0

### 88) Write a Query to display odd records from the student table?

SELECT \* FROM (SELECT \*, ROW\_NUMBER() OVER (ORDER BY student\_no) AS Row\_ID FROM student)

WHERE row\_id %2!=0

### **List of Related Microsoft Certification Courses:**