1. [Summary of the differences between clustered and non-clustered indexes](#CLUSTERED_NONCLUSTERED)
2. [**What are the types of statements in JDBC?**](#TYPES_OF_STMTS_INJDBC)
3. [DIFF\_STORED PROCEDURE\_fUNCTION](#DIFF_STOREDPROCEDURE_FUNCITON)
4. [DIFF\_PRIMARYKEY\_FOREIGN\_KEY](#DIFF_PRIMARYKEY_FOREIGN_KEY)
5. [Diff between union and union all?](#DIFF_UNION_UNION_all)
6. [Difference between truncate and drop?](#DIFF_TRUNCATE_DROP_DELETE)
7. [ISOLATION LEVELS](#isolation_levels)
8. [What is the purpose of “with check” option in views?](#check_option)
9. [What performs well “not in” or “not exists”?](#DIFF_NOTIN_NOTEXISTS)
10. [Types of LOCKS](#types_of_locks)
11. [Which performs well a join or subquery? from memory perspective?](#SUBQUERY_JOINS)
12. [How to add NOT NULL Column in a table](#HOW_TO_aDD_NOT_NULL_TABLE)
13. [**How to get the list of tables in sybase?**](http://www.geekinterview.com/question_details/45646)
14. [DELETE DUPLICAETS AND KEEP ONE IN THE TABLE](#DELETE_DUPLICATES_KEEP_ONE)
15. [FIND SECOND HIGHEST SALARY](#second_highest_salary)
16. HIGHEST SALAARY FINDING
17. Relation database e.g. [Oracle](http://javarevisited.blogspot.sg/2012/04/java-program-to-connect-oracle-database.html), [MySQL](http://javarevisited.blogspot.sg/2010/10/frequently-used-mysql-commands-part-1.html), Sybase or MSSQL
18. [Morgan Stanley](#morgan_stanely)

* What is database index and what is cluster index?

<http://www.programmerinterview.com/index.php/database-sql/clustered-vs-non-clustered-index/>

Summary of the differences between clustered and non-clustered indexes

Here’s a summary of the differences:

* A clustered index determines the order in which the rows of the table will be stored on disk – and it actually stores row level data in the leaf nodes of the index itself. A non-clustered index has no effect on which the order of the rows will be stored.
* Using a clustered index is an advantage when groups of data that can be clustered are frequently accessed by some queries. This speeds up retrieval because the data lives close to each other on disk. Also, if data is accessed in the same order as the clustered index, the retrieval will be much faster because the physical data stored on disk is sorted in the same order as the index.
* A clustered index can be a disadvantage because any time a change is made to a value of an indexed column, the subsequent possibility of re-sorting rows to maintain order is a definite performance hit.
* A table can have multiple non-clustered indexes. But, a table can have only one clustered index.
* Non clustered indexes store both a value and a pointer to the actual row that holds that value. Clustered indexes don’t need to store a pointer to the actual row because of the fact that the rows in the table are stored on disk in the same exact order as the clustered index – and the clustered index actually stores the row-level data in it’s leaf nodes.

A **database index** is a [data structure](http://en.wikipedia.org/wiki/Data_structure) that improves the speed of data retrieval operations on a [database table](http://en.wikipedia.org/wiki/Table_(database)) at the cost of additional writes and the use of more storage space to maintain the extra copy of data. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more [columns of a database table](http://en.wikipedia.org/wiki/Column_(database)), providing the basis for both rapid random [lookups](http://en.wikipedia.org/wiki/Lookup) and efficient access of ordered records.

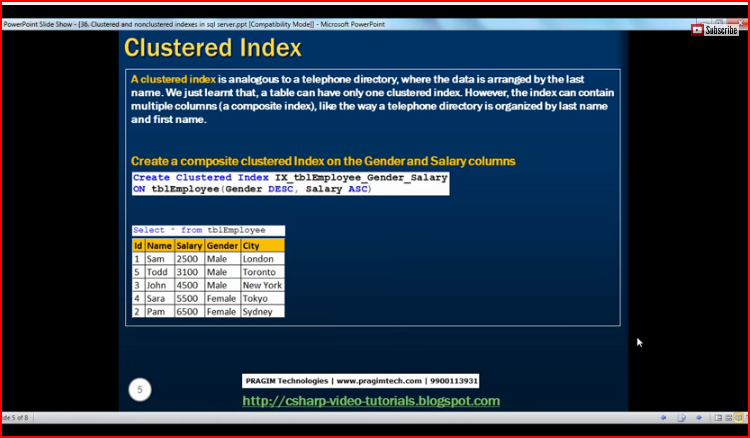
**An index is a copy of select columns of data from a table that can be searched very efficiently that also includes a low level disk block address** or direct link to the complete row of data it was copied from. Some databases extend the power of indexing by allowing indices to be created on functions or [expressions](http://en.wikipedia.org/wiki/Expression_(programming)). For example, an index could be created on upper(last\_name), which would only store the upper case versions of the last\_name field in the index. Another option sometimes supported is the use of "filtered" indices, where index entries are created only for those records that satisfy some conditional expression. A further aspect of flexibility is to permit indexing on [user-defined functions](http://en.wikipedia.org/wiki/User-defined_function), as well as expressions formed from an assortment of built-in functions.

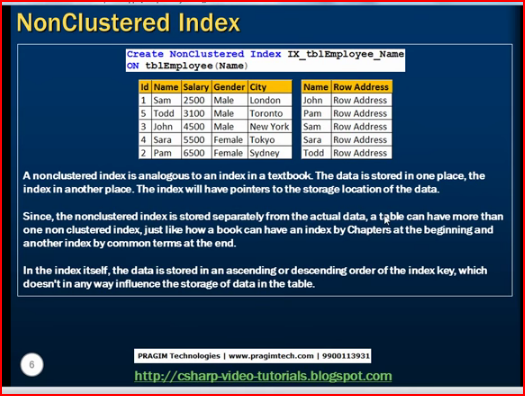
you should avoid a clustered index when; the column has low cardinality, no particular order, frequently updated, non-sequential, it is a composite of many columns.

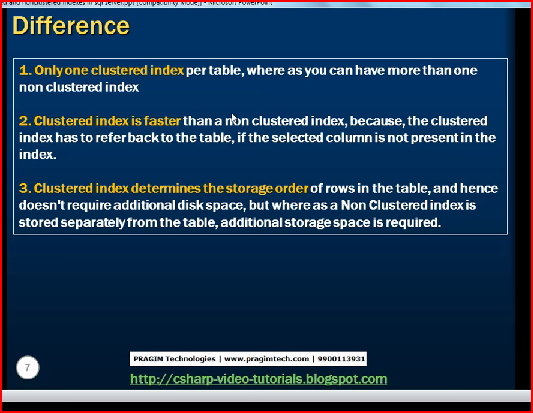
Very easy tutorial on youtube:

<https://www.youtube.com/watch?v=NGslt99VOCw>

1. When we create a tab le with primary key it automatically creates clustered index
2. Clustered indeed actually sorts data ,so only one clustered index exists in the table





1. 

# Difference between Stored Procedure and Function in SQL Server

<http://www.dotnet-tricks.com/Tutorial/sqlserver/7EDL150912-Difference-between-Stored-Procedure-and-Function-in-SQL-Server.html>

[Different types of Stored Procedure](http://www.dotnet-tricks.com/Tutorial/sqlserver/IbUO310312-Different-Types-of-SQL-Server-Stored-Procedures.html) and [Different types of Function](http://www.dotnet-tricks.com/Tutorial/sqlserver/KY3T010412-Different-Types-of-SQL-Server-Functions.html).

## Basic Difference

1. Function must return a value but in Stored Procedure it is optional( Procedure can return zero or n values).
2. Functions can have only input parameters for it whereas Procedures can have input/output parameters .
3. Function takes one input parameter it is mandatory but Stored Procedure may take o to n input parameters..
4. Functions can be called from Procedure whereas Procedures cannot be called from Function.

## Advance Difference

1. Procedure allows SELECT as well as DML(INSERT/UPDATE/DELETE) statement in it whereas Function allows only SELECT statement in it.
2. Procedures can not be utilized in a SELECT statement whereas Function can be embedded in a SELECT statement.
3. Stored Procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section whereas Function can be.
4. Functions that return tables can be treated as another rowset. This can be used in JOINs with other tables.
5. Inline Function can be though of as views that take parameters and can be used in JOINs and other Rowset operations.
6. Exception can be handled by try-catch block in a Procedure whereas try-catch block cannot be used in a Function.
7. We can go for Transaction Management in Procedure whereas we can't go in Function

<http://javadecodedquestions.blogspot.com/2011/12/>

**1) what is Index and difference with cluster index?**

### Non-clustered

The data is present in arbitrary order, but the **logical ordering** is specified by the index. The data rows may be spread throughout the table regardless of the value of the indexed column or expression. The non-clustered index tree contains the index keys in sorted order, with the leaf level of the index containing the pointer to the record (page and the row number in the data page in page-organized engines; row offset in file-organized engines).  
In a non-clustered index:

* The physical order of the rows is not the same as the index order.
* Typically created on non-primary key columns used in JOIN, WHERE, and ORDER BY clauses.

There can be more than one non-clustered index on a database table.

### Clustered

Clustering alters the data block into a certain distinct order to match the index, resulting in the row data being stored in order. Therefore, only one clustered index can be created on a given database table. Clustered indices can greatly increase overall speed of retrieval, but usually only where the data is accessed sequentially in the same or reverse order of the clustered index, or when a range of items is selected.  
Since the physical records are in this sort order on disk, the next row item in the sequence is immediately before or after the last one, and so fewer data block reads are required. The primary feature of a clustered index is therefore the ordering of the physical data rows in accordance with the index blocks that point to them. Some databases separate the data and index blocks into separate files, others put two completely different data blocks within the same physical file(s). Create an object where the physical order of rows is the same as the index order of the rows and the bottom (leaf) level of clustered index contains the actual data rows.   
Good read on index : <http://www.orafaq.com/node/1403>

**What are the types of statements in JDBC?**

           The JDBC API has 3 Interfaces, (1. Statement, 2. PreparedStatement, 3. CallableStatement ). The     key features of these are as follows:  
**Statement**

         This interface is used for executing a static SQL statement and returning the results it produces.

         The object of Statement class can be created using Connection.createStatement() method.

**PreparedStatement**

o A SQL statement is pre-compiled and stored in a PreparedStatement object.

o This object can then be used to efficiently execute this statement multiple times.

o The object of PreparedStatement class can be created using Connection.prepareStatement() method. This extends Statement interface.

**CallableStatement**

o This interface is used to execute SQL stored procedures.

o This extends PreparedStatement interface.

o The object of CallableStatement class can be created using Connection.prepareCall() method.

**What is RowSet? or What is the difference between RowSet and ResultSet? or Why do we need RowSet? or What are the advantages of using RowSet over ResultSet?**

RowSet makes it possible to use the ResultSet object as a JavaBeans component. As a consequence, a result set can, for example, be a component in a Swing application.

        o RowSet be used to make a ResultSet object scrollable and updatable. All RowSet objects are by default scrollable and updatable. If the driver and database being used do not support scrolling and/or updating of result sets, an application can populate a RowSet object implementation (e.g. JdbcRowSet) with the data of a ResultSet object and then operate on the RowSet object as if it were the ResultSet object

# [difference between primary key and unique key](http://stackoverflow.com/questions/9565996/difference-between-primary-key-and-unique-key)

Primary Key:

* Can be only one in a table
* It never allows null values
* Primary Key is unique key identifier and cannot be null and must be unique.

Unique Key:

* Can be more than one unique key in one table.
* Unique key can have null values
* It can’t be candidate key
* Unique key can be null and may not be unique.

 Primary key is a unique key.

Each table must have **at most** ONE primary key but it can have multiple unique key. A primary key is used to uniquely identify a table row. A primary key cannot be NULL since NULL is not a value.

**SYBASE:**

**RBS Sybase Interview questions**

1. How to remove duplicates from a table?

Ans: select column from table group by column having count(column) > 1

2. How do you check locks in database?

Ans: We can use the sp\_lock and sp\_familylock to see the locks are available in database.

3. How do you check processes running in Sybase?

 select \*from sysprocesses ( Here we can the cpu utilization,Engine number and Blocked processes) or sp\_who

There is the mda table monProcess and monProcessLookup in addition to sysprocesses

You can also use:  
select \* from master..sysprocesses where status =   
  
status  
-------  
sleeping   
runnable   
recv sleep   
running

4. Table a (col1, col2) Select \* from a where col1 =c and col2 =d --- how to improve perf of this query

If you want to improve the performance of this query ,we have to create index on the same.

5. Given a table .. how will u analyze the table?

We can analyze the table using the query plan of that table (sp\_showplan).  
6. Diff between union and union all?

**UNION removes duplicate records (where all columns in the results are the same), UNION ALL does not.** There is a performance hit when using UNION vs UNION ALL, since the database server must do additional work to remove the duplicate rows, but usually you do not want the duplicates (especially when developing reports)  
7. Difference between truncate and drop?

Truncate : It will truncate the data of the table.  
Drop : It will remove the data and syntax as well.

**MORGAN STANTLEY(MUMBAI)**

What is the default isolation level in sybase and what is the purpose of using isolation levels.

Default isolation level is 1. isolation levels specifies the kinds of interactions that are not permitted while concurrent transactions are executing—that is, whether transactions are isolated from each other, or if they can read or update information in use by another transaction. **Sybase supports 4 isolation levels level 0 (read uncommitted), level 1( read committed), level 2(repeatable read) and level 3( serializable read)**

What is the purpose of “with check” option in views?

with check option restricts the rows that can be updated or inserted on the where clause  
  
create view *vw\_ca\_authors* as  
select au\_id, au\_lname, au\_fname, phone, state  
from authors  
where state = “CA”  
with check option

-- can only insert or update to “CA” so this insert fails  
insert vw\_ca\_authors values (“111-222-3333”, “Smith”, “John”, “453-2343”, “NY”)

**Any view or stored procedure that references the dropped table must be explicitly dropped by using DROP VIEW or DROP PROCEDURE.**

To report the dependencies on a table, use sys.dm\_sql\_referencing\_entities. (For SQL database)

1. What performs well “not in” or “not exists”?

When using “NOT IN”, the query performs nested full table scans, whereas for “NOT EXISTS”, query can use an index within the sub-query

1. Diff .Clustered and non-clustered and when to create them. Number of clustered / non – clustered indexes that can be created on a specific table?

There can be only 1 clustered index and upto 249 non clustered indexes per table. in clustered indexes physical and logical data order is same and there are fewer index levels and hence faster compared to non clustered index. In non clustered physical and logical order is not same and there is a leaf page to which data page is linked in non clustered index.

Usually clustered indexes are better suited for range queries and non clustered for point search queries

1. Types of locks in sybase, Is shared on shared lock, shared on exclusive, exclusive on exclusive lock possible?

shared on shared and shared on exclusive are possible but exclusive on exclusive is not.

1. How many parameters can a Stored Procedure return?

maximum number of parameters that a stored procedure can have is 255.

1. Which performs well a join or subquery? from memory perspective?

If memory is ample then Joins are preferable. Join has a better performance over sub-queries as subqueries involves creation of intermediate tables and more I/O.

What are the levels of groups in Sybase?  
There is no levels in sybase. By default only public group is created.

4) How do we check if the indexes on a particular table are not corrupt OR How to check the indexes are consistent ?   
  
Run dbcc indexalloc

How to [Add not null column in a table](http://www.geekinterview.com/question_details/69541)

First add the column as NULL. Then populate the column with non-null data. Then modify the column to NOT NULL. You may need to set dboption "select into" to True.

What is the use of 'with check' option in views?

The with check option flag prevents insertion or updating of rows that will not meet the view criteria. e.g.: create view cal\_publishers\_chk as ..

**[How to get the list of tables in sybase?](http://www.geekinterview.com/question_details/45646)**

Sp\_help - lists all objects (including tables and views)   
**List if Views :** select name from sysobjects where type = "V"

**Select name from sysobjects where type="U"** This gives the list of user defined tables in the current database.

[**Given a table which contains some rows with duplicate keys, how would you remove the duplicates?**](http://www.geekinterview.com/question_details/33103)

Delete from table1 where rowid not in (select max(rowid) from table1 group by dup\_col

[**Db space**](http://www.geekinterview.com/question_details/76797)

Sp\_helpdb gives information about memory used and available.sp\_spaceused - gives space used..

[**Query performance tuning**](http://www.geekinterview.com/question_details/82790)

Do a set showplan oncheck each query and see if any table scan is used....if yes, check if any index is present on tables...if index is not present try adding it....if index is present, determine what...

[**Which one is faster count(\*) or count(1) ?**](http://www.geekinterview.com/question_details/32999)

Count(1) is faster than count(\*)

[**What is the major differences between Oracle & sybase**](http://www.geekinterview.com/question_details/39984)

 while Sybase does not have "Object Tables" it can hold BLOBS in columns in tables. These can be of datatype TEXT or IMAGE. Sybase has the concept of Temporary tables in the form of ...

The dialect of SQL supported by Microsoft Sybase ASE is called Transact-SQL (T-SQL). The dialect of SQL supported by Oracle 9i Database is called PL/SQL. PL/SQL is more powerful language than T-SQL.

the main difference between Sybase and [Oraclehttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/39984) is that   
in Oracle, it manages all of the data and database related functions within ONE   
big Oracle Instance. Sybase manages all of the [datahttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/39984) and database related   
functions thru a collection of system [databaseshttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/39984) and user databases all in ONE   
Sybase server. Each database within a [Sybasehttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/question_details/39984) server can communicate with one   
another, but they are compartmentalized. Where as in Oracle they are all mingle   
together in one space.

[**A column name is given to you. You have to find out which table has that column. Database has 230 tables. For example: empid (column name) for this how do you find out which table has the column (empid)...**](http://www.geekinterview.com/question_details/51804)

SELECT DISTINCT o.name FROM sysobjects o, syscolumns c WHERE o.id=c.id AND c.name='empid'

  --- or you can go for c.name LIKE "%empid%"

How to find the 2 minimum salaries in a table?

select min(salary) from employee where salary >(select min(salary) from employee)

[**What are the data types supported by sybase?**](http://www.geekinterview.com/question_details/45650)

The [data typeshttp://images.intellitxt.com/ast/adTypes/icon1.png](http://www.geekinterview.com/Interview-Questions/Database/Sybase/page3) supported by Sybase are integer, smallint, tinyint, float, real, double precision, smallmoney, money, decimal, char(n), varchar(n), text, image, datetime, timestamp, timestamp, varbinary(n), binary(n) and bit.

| **Table 11-2: Storage sizes for Adaptive Server datatypes** | |
| --- | --- |
| **Datatype** | **Size** |
| *char* | Defined size |
| *nchar* | Defined size \* *@@ncharsize* |
| *unichar* | n\**@@unicharsize (@@unicharsize equals 2)* |
| *univarchar* | the actual number of characters\**@@unicharsize* |
| *varchar* | Actual number of characters |
| *nvarchar* | Actual number of characters \* *@@ncharsize* |
| *binary* | Defined size |
| *varbinary* | Data size |
| *int* | 4 |
| *smallint* | 2 |
| *tinyint* | 1 |
| *float* | 4 or 8, depending on precision |
| *double precision* | 8 |
| *real* | 4 |
| *numeric* | 2–17, depending on precision and scale |
| *decimal* | 2–17, depending on precision and scale |
| *money* | 8 |
| *smallmoney* | 4 |
| *datetime* | 8 |
| *smalldatetime* | 4 |
| *bit* | 1 |
| *text* | 16 bytes + 2K \* number of pages used |
| *image* | 16 bytes + 2K \* number of pages used |
| *timestamp* | 8 |

Oracle QUESTIONS:

|  |  |
| --- | --- |
| What are the datatypes a available in PL/SQL? | |
|  | Some scalar data types such as  NUMBER, VARCHAR2, DATE, CHAR, LONG, BOOLEAN. Some composite data types such as RECORD & TABLE. | |

* 1. **Cursor :** Oracle creates a memory area, known as context area, for processing an SQL statement, which contains all information needed for processing the statement, for example, number of rows processed, etc.

<http://www.tutorialspoint.com/plsql/plsql_cursors.htm>

A cursor holds the rows (one or more) returned by a SQL statement

You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors:.

|  |  |
| --- | --- |
| What is a cursor? Why Cursor is required? | |
|  | Cursor is a named private SQL area from where information can be accessed. Cursors are required to process rows individually for queries returning multiple rows. | |

* Implicit cursors – no control, automatically created by system.
* Explicit cursors
  + The syntax for creating an explicit cursor is :
* CURSOR c\_customers IS
* SELECT id, name, address FROM customers;
  + OPEN c\_customers;
  + FETCH c\_customers INTO c\_id, c\_name, c\_addr;
* CLOSE c\_customers;

ORACLE PROCEDURE

|  |  |
| --- | --- |
| Give the structure of the procedure? | |
|  | PROCEDURE name (parameter list.....) is local variable declarations  BEGIN Executable statements. Exception. exception handlers  end;  ORACLE FUNCTION  .   |  |  | | --- | --- | | Give the structure of the function? | | |  | FUNCTION name (argument list .....) Return datatype is local variable declarations Begin executable statements Exception execution handlers End; | | | |

1. view

In **SQL**, a **view** is a virtual table based on the result-set of an **SQL**statement. A **view** contains rows and columns, just like a real table. The fields in a **view** are fields from one or more real tables in the database.

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

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.

The basic CREATE VIEW syntax is as follows:

CREATE VIEW view\_name AS

SELECT column1, column2.....

FROM table\_name

WHERE [condition];

The following is an example of creating same view CUSTOMERS\_VIEW with the WITH CHECK OPTION:

CREATE VIEW CUSTOMERS\_VIEW AS

SELECT name, age

FROM CUSTOMERS

WHERE age IS NOT NULL

WITH CHECK OPTION;

The WITH CHECK OPTION in this case should deny the entry of any NULL values in the view's AGE column, because the view is defined by data that does not have a NULL value in the AGE column.

DROP VIEW view\_name;

Updating a View:

A view can be updated under certain conditions:

* The SELECT clause may not contain the keyword DISTINCT.
* The SELECT clause may not contain summary functions.
* The SELECT clause may not contain set functions.
* The SELECT clause may not contain set operators.
* The SELECT clause may not contain an ORDER BY clause.
* The FROM clause may not contain multiple tables.
* The WHERE clause may not contain subqueries.
* The query may not contain GROUP BY or HAVING.
* Calculated columns may not be updated.
* All NOT NULL columns from the base table must be included in the view in order for the INSERT query to function.
* UPDATE CUSTOMERS\_VIEW
* SET AGE = 35
* WHERE name='Ramesh';

DISADVANTAGES

1. When table is dropped or modified, view becomes inactive, it depends on the table objects.
2. Not all the time we can perform DML statements, as normally views are made for complex query and depends on more than one table. So there is more possibilities of violating your database constrains while performing DML statements.
3. As views are normally used for a complex static query, not all the times we can have same situation to use that static query. For example; If you are querying over view, then it looks like you save time , but if you are looking for few information from view, then you may face performance degradation problem.
4. when table is not there view will not work.
5. dml is not possible if that is more than one table.
6. ~~it is also database object so it will occupy the space~~. as views are just stored statements, so it does not occupy space, except storing the statement
7. When table is dropped view becomes inactive.. it depends on the table objects.
8. Querying from view takes more time than directly querying from the table

**trigger**

A **trigger** is a special kind of stored procedure that automatically executes when an event occurs in the database server. DML **triggers** execute when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

Sybase syntax:

create trigger [*owner*.]*trigger\_name*

on [*owner*.]*table\_name*

{for {insert , update} | instead of {insert, update, delete}}

[as

[if update (*column\_name*)

[{and | or} update (*column\_name*)]...]

*SQL\_statements*

[if update (*column\_name*)

[{and | or} update (*column\_name*)]...

*SQL\_statements*]...]

#### Example 1

Prints a message when anyone tries to add data or change data in the *titles* table:

create trigger reminder

on titles

for insert, update as

print "Don't forget to print a report for accounting."

#### Example 2

Prevents insertion of a new row into *titleauthor* if there is no corresponding *title\_id* in the *titles* table:

create trigger t1

on titleauthor

for insert as

if (select count (\*)

    from titles, inserted

    where titles.title\_id = inserted.title\_id) = 0

begin

print "Please put the book's title\_id in the

        titles table first."

rollback transaction

end

Oracle Syntax:

* 1. What is normalization in sQL

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process:

* + 1. eliminating redundant data (for example, storing the same data in more than one [table](http://databases.about.com/library/glossary/bldef-table.htm) )
    2. ensuring [data dependencies](http://databases.about.com/od/specificproducts/a/Database-Dependency.htm) make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

In practical applications, you'll often see [1NF](http://databases.about.com/od/specificproducts/l/aa1nf.htm) , [2NF](http://databases.about.com/od/specificproducts/a/2nf.htm) , and [3NF](http://databases.about.com/od/specificproducts/l/aa3nf.htm) along with the occasional 4NF.

### First normal form (1NF) sets the very basic rules for an organized database:

* Eliminate duplicative [columns](http://databases.about.com/library/glossary/bldef-column.htm) from the same table.
* Create separate tables for each group of related data and identify each [row](http://databases.about.com/library/glossary/bldef-row.htm) with a unique column or set of columns (the [primary key](http://databases.about.com/library/glossary/bldef-primarykey.htm)).

### Second Normal Form (2NF)

Second normal form (2NF) further addresses the concept of removing duplicative data:

* Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
* Create relationships between these new tables and their predecessors through the use of [foreign keys](http://databases.about.com/library/glossary/bldef-foreignkey.htm).

### Third Normal Form (3NF)

Third normal form (3NF) goes one large step further:

* Meet all the requirements of the second normal form.
* Remove columns that are not [dependent](http://databases.about.com/od/specificproducts/a/Database-Dependency.htm) upon the primary key.

### Boyce-Codd Normal Form (BCNF or 3.5NF)

The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form", adds one more requirement:

* Meet all the requirements of the third normal form.
* Every determinant must be a [candidate key.](http://databases.about.com/cs/specificproducts/g/candidate.htm)

[For more details, read](http://databases.about.com/cs/specificproducts/g/candidate.htm)[Putting your Database in Boyce Codd Normal Form](http://databases.about.com/cs/specificproducts/g/bcnf.htm)

### Fourth Normal Form (4NF)

Finally, fourth normal form (4NF) has one additional requirement:

* Meet all the requirements of the third normal form.
* A relation is in 4NF if it has no [multi-valued dependencies](http://databases.about.com/od/specificproducts/g/multivalued-dependency.htm).

Remember, these normalization guidelines are cumulative. For a database to be in 2NF, it must first fulfill all the criteria of a 1NF database.

# Candidate Key

**Definition:**A candidate key is a combination of attributes that can be uniquely used to identify a database record without any extraneous data. Each table may have one or more candidate keys. One of these candidate keys is selected as the table primary key.

# [What is the difference between a candidate key and a primary key?](http://stackoverflow.com/questions/12813363/what-is-the-difference-between-a-candidate-key-and-a-primary-key)

Candidate Key – A Candidate Key can be any column or a combination of columns that can **qualify as unique key in database**. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.

Primary Key – A Primary Key is a column or a combination of columns that **uniquely identify a record**. Only one Candidate Key can be Primary Key.

[**More on this link with example**](http://blog.sqlauthority.com/2009/10/22/sql-server-difference-between-candidate-keys-and-primary-key-2/)

### Difference between primary key vs Candidate Key in table - SQL database

<http://java67.blogspot.com/2012/10/difference-between-primary-key-vs-candidate-keys-sql-database.html>

1) Both Primary and Candidate keys can uniquely identify records in a table on database.

 2) Both Primary and Candidate keys are has constraints UNIQUE and NOT NULL.

 3) Primary key or Candidate keys can be either single column or combination of multiple columns in a table.

Now from interview point of view here is difference between Candidate key and primary key in SQL table on point format for easy to remember :

1) There can be multiple Candidate keys in a table in relation database e.g. [Oracle](http://javarevisited.blogspot.sg/2012/04/java-program-to-connect-oracle-database.html), [MySQL](http://javarevisited.blogspot.sg/2010/10/frequently-used-mysql-commands-part-1.html), Sybase or MSSQL but only one primary key is permitted.

2) An example of Primary key and Candidate key can be ID and SSN number in a Employee table, Since both can identify each employee uniquely they are candidate key and any one can become primary key. Now if you have to choose between them as primary key, I will go ID as [primary key](http://java67.blogspot.sg/2012/10/difference-between-primary-vs-unique-key-table-sql.html) because SSN is sensitive information and may not be allow/not safe to use as String in queries as frequently as ID. Second reason of choosing ID over SSN as primary key can be use of ID as primary tracking ID within organization and its frequent use all over the place. Once you choose a primary key, All candidate key  are like unique keys.

That's all on difference between Primary key and Candidate key in a table. If you understand election well than you can think primary key as elected member among all candidate keys.

# [What are database constraints really?](http://stackoverflow.com/questions/2570756/what-are-database-constraints-really)

Constraints are part of a database schema definition.

A constraint is usually associated with a table and is created with a CREATE CONSTRAINT or CREATE ASSERTION SQL statement.

They define certain properties that data in a database must comply with. They can apply to a column, a whole table, more than one table or an entire schema. A reliable database system ensures that constraints hold at all times (except possibly inside a transaction, for so called deferred constraints).

Common kinds of constraints are:

* **not null** - value in a column must not be NULL
* **unique** - value(s) in specified column(s) must be unique for each row in a table
* **primary key** - value(s) in specified column(s) must be unique for each row in a table and not be NULL; normally each table in a database should have a primary key - it is used to identify individual records
* **foreign key** - value(s) in specified column(s) must reference an existing record in another table (via its primary key or some other unique constraint)
* **check** - an expression is specified, which must evaluate to true for constraint to be satisfied

Examples:

**SQL Server / Oracle / MS Access:**

CREATE TABLE Persons  
(  
P\_Id int NOT NULL PRIMARY KEY,  
LastName varchar(255) NOT NULL,  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255)  
)

Sybase

CREATE TABLE skill (

skill\_id INTEGER NOT NULL,

skill\_name CHAR( 20 ) NOT NULL,

skill\_type CHAR( 20 ) NOT NULL,

primary key( skill\_id )

)

# Sybase  check constraints

For example, this statement ensures that only certain values can be entered for the *pub\_id* column:

create table my\_new\_publishers

(pub\_id      char(4)

    check (pub\_id in ("1389", "0736", "0877",

             "1622", "1756")

        or pub\_id like "99[0-9][0-9]"),

pub\_name     varchar(40),

city         varchar(20),

state        char(2))

You can still insert NULL into that column. The column definition overrides the check constraint because the following expression always evaluates to true:

col\_name != null

# [What is the difference between count(0), count(1).. and count(\*) in mySQL/SQL?](http://stackoverflow.com/questions/18291036/what-is-the-difference-between-count0-count1-and-count-in-mysql-sql)

Nothing really, unless you specify a field in a table or an expression within parantheses instead of constant values or \*

Let me give you a detailed answer. Count will give you non-null record number of given field. Say you have a table named A

select 1 from A

select 0 from A

select \* from A

will all return same number of records, that is the number of rows in table A. Still the output is different. If there are 3 records in table. With X and Y as field names

select 1 from A will give you

1

1

1

select 0 from A will give you

0

0

0

select \* from A will give you ( assume two columns X and Y is in the table )

X Y

-- --

value1 value1

value2 (null)

value3 (null)

So, all three queries return the same number. Unless you use

select count(Y) from A

since there is only one non-null value you will get 1 as output

Use count(1) instead of count(\*)

In days gone by, though, COUNT(1) (or whatever constant you chose) was sometimes recommended over COUNT(\*) because poor query optimisation code would make the database retrieve all of the field data prior to running the count. COUNT(1) was therefore faster, but it shouldn't matter now.

**Sybase**

# Need to substring a text data type in sybase

Sybase provide the SUBSTRING function with this sintax:  
SUBSTRING( string-expression, start [, length ] )

Convert text to date in Sybase

**CONVERT** ( *data-type*, *expression* [ , *format-style* ] )

EX:

select \* from data where dateVal < convert(datetime, '01/01/2008', 103)

Returns the current system date and time.

### Description

Returns the current system date and time.

### Syntax

getdate()

### Parameters

None.

### Examples

#### Example 1

Assumes a current date of November 25, 1995, 10:32 a.m.:

select getdate()

Nov 25 1995 10:32AM

#### Example 2

Assumes a current date of November:

select datepart(month, getdate())

11

#### Example 3

Assumes a current date of November:

select datename(month, getdate())

November

### Usage

# [Sybase: How to get the current date in mm/dd/yyyy format?](http://stackoverflow.com/questions/19070124/sybase-how-to-get-the-current-date-in-mm-dd-yyyy-format)

select convert(varchar, date\_column, 101) from the\_table

Sybase functions:

|  |  |
| --- | --- |
| String function | Parameters |
| TRIM | ( *string-expr* ) |
| UCASE | ( *string-expr* ) |
| UPPER | ( *string-expr* ) |
|  |  |
| SUBSTRING | ( string-expr, integer-expr [ , integer-expr ] ) |

<http://databases.about.com/od/specificproducts/a/normalization.htm>

**What is Normalization?**

Normalization is the process of efficiently organizing data in a database.

There are two goals of the normalization process:

1. Eliminating redundant data (for example, storing the same data in more than one [table](http://databases.about.com/library/glossary/bldef-table.htm))
2. and ensuring [data dependencies](http://databases.about.com/od/specificproducts/a/Database-Dependency.htm) make sense (only storing related data in a table).

Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

### The Normal Forms

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as [first normal form](http://databases.about.com/od/specificproducts/l/aa1nf.htm) or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see [1NF](http://databases.about.com/od/specificproducts/l/aa1nf.htm), [2NF](http://databases.about.com/od/specificproducts/a/2nf.htm), and[3NF](http://databases.about.com/od/specificproducts/l/aa3nf.htm) along with the occasional 4NF. Fifth normal form is very rarely seen and won't be discussed in this article.

**First Normal Form (1NF)**

First normal form (1NF) sets the very basic rules for an organized database:

* Eliminate duplicative [columns](http://databases.about.com/library/glossary/bldef-column.htm) from the same table.
* Create separate tables for each group of related data and identify each [row](http://databases.about.com/library/glossary/bldef-row.htm) with a unique column or set of columns (the [primary key](http://databases.about.com/library/glossary/bldef-primarykey.htm)).

**Second Normal Form (2NF)**

Second normal form (2NF) further addresses the concept of removing duplicative data:

* Meet all the requirements of the first normal form.
* Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
* Create relationships between these new tables and their predecessors through the use of [foreign keys](http://databases.about.com/library/glossary/bldef-foreignkey.htm).

**Third Normal Form (3NF)**

Third normal form (3NF) goes one large step further:

* Meet all the requirements of the second normal form.
* Remove columns that are not [dependent](http://databases.about.com/od/specificproducts/a/Database-Dependency.htm) upon the primary key.

For more details, read [**Putting your Database in Third Normal Form**](http://databases.about.com/od/specificproducts/l/aa3nf.htm)

**Boyce-Codd Normal Form (BCNF or 3.5NF)**

The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form", adds one more requirement:

* Meet all the requirements of the third normal form.
* Every determinant must be a [candidate key.](http://databases.about.com/cs/specificproducts/g/candidate.htm)

<http://stackoverflow.com/questions/17946221/sql-join-and-different-types-of-joins>

# What is SQL JOIN ?

SQL JOIN is a method to retrieve data from two or more database tables.

# What are the different SQL JOINs ?

There are a total of five JOINs. They are :

1. JOIN or INNER JOIN

2. OUTER JOIN

2.1 LEFT OUTER JOIN or LEFT JOIN

2.2 RIGHT OUTER JOIN or RIGHT JOIN

2.3 FULL OUTER JOIN or FULL JOIN

3. NATURAL JOIN

4. CROSS JOIN

5. SELF JOIN

## 1. JOIN or INNER JOIN :

In this kind of a JOIN, we get all records that match the condition in both the tables, and records in both the tables that do not match are not reported.

In other words, INNER JOIN is based on the single fact that : ONLY the matching entries in BOTH the tables SHOULD be listed.

Note that a JOIN without any other JOIN keywords (like INNER, OUTER, LEFT, etc) is an INNER JOIN. In other words, INNER JOIN is a Syntactic sugar for JOIN (see : [Difference between JOIN and INNER JOIN](http://stackoverflow.com/questions/565620/difference-between-join-and-inner-join)).

## 2. OUTER JOIN :

Outer Join retrieves

Either, the matched rows from one table and all rows in the other table Or, all rows in all tables (it doesn't matter whether or not there is a match). n

There are three kinds of Outer Join :

**2.1 LEFT OUTER JOIN or LEFT JOIN**

This join returns all the rows from the left table in conjunction with the matching rows from the right table. If there are no columns matching in the right table, it returns NULL values.

**2.2 RIGHT OUTER JOIN or RIGHT JOIN**

This join returns all the rows from the right table in conjunction with the matching rows from the left table. If there are no columns matching in the left table, it returns NULL values.

**2.3 FULL OUTER JOIN or FULL JOIN**

**This join combines left outer join and right outer join. It returns row from either table when the conditions are met and returns null value when there is no match.**

In other words, OUTER JOIN is based on the fact that : ONLY the matching entries in ONE OF the tables (RIGHT or LEFT) or BOTH of the tables(FULL) SHOULD be listed.

Note that OUTER JOIN is a loosened form of INNER JOIN.

## 3. NATURAL JOIN :

It is based on the two conditions :

1. the JOIN is made on all the columns with the same name for equality.
2. Removes duplicate columns from the result.

This seems to be more of theoretical in nature and as a result (probably) most DBMS don't even bother supporting this.

## 4. CROSS JOIN :

It is the Cartesian product of the two tables involved. The result of a CROSS JOIN will not make sense in most of the situations. Moreover, we wont need this at all (or needs the least, to be precise).

## 5. SELF JOIN :

It is not a different form of JOIN, rather it is a JOIN (INNER, OUTER, etc) of a table to itself.

# JOINs based on Operators

Depending on the operator used for a JOIN clause, there can be two types of JOINs. They are

1. Equi JOIN
2. Theta JOIN

## 1. Equi JOIN :

For whatever JOIN type (INNER, OUTER, etc), if we use ONLY the equality operator (=), then we say that the JOIN is an Equi JOIN.

## 2. Theta JOIN :

This is same as Equi JOIN but it allows all other operators like >, <, >= etc.

Many consider both Equi JOIN and Theta JOIN similar to INNER, OUTER etc JOINs. But I strongly believe that its a mistake and makes the ideas vague. Because INNER JOIN, OUTER JOIN etc are all connected with the tables and their data where as Equi JOIN and Theta JOIN are only connected with the operators we use in the former.

Again, there are many who consider NATURAL JOIN as some sort of "peculiar" Equi JOIN. In fact, it is true, because of the first condition I mentioned for NATURAL JOIN. However, we dont have to restrict that simply to NATURAL JOINs alone. INNER JOINs, OUTER JOINs etc could be an Equi JOIN too.

*Candidate Key* is a column or combination of columns, which can be a *Primary key* for the Table.  
There may be multiple *Candidate Keys* In a Table.  
   
Any *Candidate Key* can be a *Primary Key*.  
   
*Unique Key* is a column which has unique value for each row, but it also allow nulls (*Primary Key* doesn't).  
   
Any confusions, feel free to ask.

# [delete duplicate rows and need to keep one from all of them in mysql](http://stackoverflow.com/questions/15548655/delete-duplicate-rows-and-need-to-keep-one-from-all-of-them-in-mysql)

I want to delete duplicate rows based on two columns but need to keep 1 row all of them.

Duplicate rows can be more than two rows like,

ID NAME PHONE

-- ---- ----

1 NIL 1234

2 NIL 1234

3 NIL 1234

4 MES 5989

I want to delete any of 2 rows from above 3 and keep 1 row.

DELETE a

FROM tableA a

LEFT JOIN

(

SELECT MIN(ID) ID, Name, Phone

FROM TableA

GROUP BY Name, Phone

) b ON a.ID = b.ID AND

a.NAme = b.Name AND

a.Phone = b.Phone

WHERE b.ID IS NULL

OTHER WAY OF DOING

MYSQL[orts JOINs in DELETE statements](http://dev.mysql.com/doc/refman/5.0/en/delete.html). If you want to keep the first of the duplicates:

DELETE a

FROM MYVIEWS a

JOIN (SELECT MIN(t.a1) AS min\_a1, t.k1, t.k2, t.k3

FROM MYVIEWS t

GROUP BY t.k1, t.k2, t.k3

HAVING COUNT(\*) > 1) b ON b.k1 = a.k1

AND b.k2 = a.k2

AND b.k3 = a.k3

AND b.min\_a1 != a.a1

If you want to keep the last of the duplicates:

DELETE a

FROM MYVIEWS a

JOIN (SELECT MAX(t.a1) AS max\_a1, t.k1, t.k2, t.k3

FROM MYVIEWS t

GROUP BY t.k1, t.k2, t.k3

HAVING COUNT(\*) > 1) b ON b.k1 = a.k1

AND b.k2 = a.k2

AND b.k3 = a.k3

AND b.max\_a1 != a.a1

know how to find 2nd highest salary

SELECT MAX(Salary) FROM Employee

WHERE Salary NOT IN (SELECT MAX(Salary) FROM Employee )

How to find top three highest salary in emp table in oracle?

SELECT \*FROM

(

SELECT \*FROM emp

ORDER BY Salary desc

)

WHERE rownum <= 3

ORDER BY Salary ;

Morgan Stanley

1. Difference between minus, union and intersection

Ans:The SQL **INTERSECT operator** is used to return the results of 2 or more SELECT statements. However, it only returns the rows selected by all queries. If a record exists in one query and not in the other, it will be omitted from the INTERSECT results.

The SQL **MINUS operator** is used to return all rows in the first SELECT statement that are not returned in the second SELECT statement.

         What is save point, have you used save point.

A **SAVEPOINT** is a marker within a transaction that allows for a partial rollback. As changes are made in a transaction, we can create **SAVEPOINTs** to mark different points within the transaction. If we encounter an error, we can rollback to a **SAVEPOINT** or all the way back to the beginning of the transaction.

         Case statement in DB. How to use that.

         Suppose a customer has n accounts and accounts has n transactions. How you will design the tables.

         If person able to design above, they ask query to find out customer who is having maximum number of transaction

         Employee table having Employee Id, Employee Name and Employee Salary. Find out employee who is having second highest salary