

SQL interview questions and answers

1. **What are two methods of retrieving SQL?**
2. **What cursor type do you use to retrieve multiple recordsets?**
3. **What is the difference between a "where" clause and a "having" clause?** - "Where" is a kind of restriction statement. You use where clause to restrict all the data from DB. Where clause is using before result retrieving. But Having clause is using after retrieving the data. Having clause is a kind of filtering command.
4. **What is the basic form of a SQL statement to read data out of a table?** The basic form to read data out of table is 'SELECT * FROM table_name;'. An answer: 'SELECT * FROM table_name WHERE xyz= 'whatever';' cannot be called basic form because of WHERE clause.
5. **What structure can you implement for the database to speed up table reads?** - Follow the rules of DB tuning we have to: 1] properly use indexes (different types of indexes) 2] properly locate different DB objects across different tablespaces, files and so on. 3] create a special space (tablespace) to locate some of the data with special datatype (for example CLOB, LOB and ...)
6. **What are the tradeoffs with having indexes?** - 1. Faster selects, slower updates. 2. Extra storage space to store indexes. Updates are slower because in addition to updating the table you have to update the index.
7. **What is a "join"?** - 'join' used to connect two or more tables logically with or without common field.
8. **What is "normalization"? "Denormalization"? Why do you sometimes want to denormalize?** - Normalizing data means eliminating redundant information from a table and organizing the data so that future changes to the table are easier. Denormalization means allowing redundancy in a table. The main benefit of denormalization is improved performance with simplified data retrieval and manipulation. This is done by reduction in the number of joins needed for data processing.
9. **What is a "constraint"?** - A constraint allows you to apply simple referential integrity checks to a table. There are four primary types of constraints that are currently supported by SQL Server: PRIMARY/UNIQUE - enforces uniqueness of a particular table column. DEFAULT - specifies a default value for a column in case an insert operation does not provide one. FOREIGN KEY - validates that every value in a column exists in a column of another table. CHECK - checks that every value stored in a column is in some specified list. Each type of constraint performs a specific type of action. Default is not a constraint. NOT NULL is one more constraint which does not allow values in the specific column to be null. And also it the only constraint which is not a table level constraint.
10. **What types of index data structures can you have?** - An index helps to faster search values in tables. The three most commonly used index-types are: - B-Tree: builds a tree of possible values with a list of row IDs that have the leaf value. Needs a lot of space and is the default index type for most databases. - Bitmap: string of bits for each possible value of the column. Each bit string has one bit for each row. Needs only few space and is very fast. (however, domain of value cannot be large, e.g. SEX(m,f); degree(BS,MS,PHD) - Hash: A hashing algorithm is used to assign a set of characters to

represent a text string such as a composite of keys or partial keys, and compresses the underlying data. Takes longer to build and is supported by relatively few databases.

11. **What is a "primary key"?** - A PRIMARY INDEX or PRIMARY KEY is something which comes mainly from database theory. From its behavior is almost the same as an UNIQUE INDEX, i.e. there may only be one of each value in this column. If you call such an INDEX PRIMARY instead of UNIQUE, you say something about your table design, which I am not able to explain in few words. Primary Key is a type of a constraint enforcing uniqueness and data integrity for each row of a table. All columns participating in a primary key constraint must possess the NOT NULL property.
12. **What is a "functional dependency"? How does it relate to database table design?** - Functional dependency relates to how one object depends upon the other in the database. for example, procedure/function sp2 may be called by procedure sp1. Then we say that sp1 has functional dependency on sp2.
13. **What is a "trigger"?** - Triggers are stored procedures created in order to enforce integrity rules in a database. A trigger is executed every time a data-modification operation occurs (i.e., insert, update or delete). Triggers are executed automatically on occurrence of one of the data-modification operations. A trigger is a database object directly associated with a particular table. It fires whenever a specific statement/type of statement is issued against that table. The types of statements are insert,update,delete and query statements. Basically, trigger is a set of SQL statements A trigger is a solution to the restrictions of a constraint. For instance: 1.A database column cannot carry PSEUDO columns as criteria where a trigger can. 2. A database constraint cannot refer old and new values for a row where a trigger can.
14. **Why can a "group by" or "order by" clause be expensive to process?** - Processing of "group by" or "order by" clause often requires creation of Temporary tables to process the results of the query. Which depending of the result set can be very expensive.
15. **What is "index covering" of a query?** - Index covering means that "Data can be found only using indexes, without touching the tables"
16. **What types of join algorithms can you have?**
17. **What is a SQL view?** - An output of a query can be stored as a view. View acts like small table which meets our criterion. View is a precompiled SQL query which is used to select data from one or more tables. A view is like a table but it doesn't physically take any space. View is a good way to present data in a particular format if you use that query quite often. View can also be used to restrict users from accessing the tables directly.