Database (dB) is a storage space for content / information (data); It is an organized collection of data;

♣ Relational databases use tables to store information and specify the relations between those tables;

♣ Schema is a logical container for database objects (tables, views, triggers) that user creates;

♣ Table is a container for data elements and relations. It’s using columns (fields) and rows (records);

♣ Field is a database storage simplest unit (table column);

♣ Record is a row and it represents a single structured data set in table;

♣ Query is your request to the database to retrieve information;

♣ Report is the returned information to the specified query;

♣ DBMS (Database Management System) is a software that controls the organization, storage, retrieval, security and integrity of data in dB;

♣ Popular Databases: ORACLE, Sybase, MySQL, DB2, etc.

♣ PRIMARY KEY (PK) is a unique identifier of every record in a table;

♣ FOREIGN KEY (FK) is a column (or combination of columns) that is used to establish a relationship between the tables; Foreign key is usually not unique (one-to-many relation) and shall always point to a primary key.

♣ NORMALIZATION is the dB process of organizing the fields and tables of a relationship database to minimize redundancy and dependency. Divide large tables into smaller tables and defining relationships between them;

♣ SQL stands for Structured Query Language. It is a database computer language, designed to retrieve and manage data, create and modify dB schema, etc.

♣ SQL Commands:

♣ Create, modify, delete, and view tables (CREATE, ALTER, DROP (delete table), DESC)

♣ Populate tables with data (INSERT, DELETE, TRUNCATE (delete data, but not table), etc)

♣ Query the table(s) data (SELECT) CREATE - creates a table (index, view) in a database.

CREATE TABLE table1 (ID int, Last\_Name varchar(255))

ALTER - used to add, delete, or modify columns in an existing table.

ALTER TABLE table1 ADD column1 varchar(255)

ALTER TABLE table1 DROP COLUMN column1

ALTER TABLE table1 MODIFY column1 varchar(255)

DROP – deletes a table (index, view).

DROP TABLE table1

TRUNCATE - deletes data in the table, but not table itself

TRUNCATE TABLE table1 INSERT - inserts record into a table (don’t violate constraints)

INSERT into table1 VALUES (value1, value2, etc.)

UPDATE - modifies record in a table (no primary keys update)

UPDATE table1 SET column1 = ‘Smith’ WHERE column1 = ‘Allins’

DELETE - deletes record from a table (all records, if there is no WHERE)

DELETE FROM table1 [WHERE column1 = ‘Allins’]

SELECT - retrieves data from one or more dB tables or views.

Most often used statement DDL (Data Definition Language) statements are defining database schema:

♣ CREATE - to create objects in the database (TABLE, VIEW, INDEX)

♣ ALTER - alters the structure of the database (TABLE, VIEW, INDEX)

♣ DROP - delete objects from the database (TABLE, VIEW, INDEX)

♣ TRUNCATE - remove all records from a table

♣ COMMENT - add comments to the data dictionary

♣ RENAME - rename an object

DML (Data Manipulation Language) statements are managing data within schema:

♣ SELECT - retrieve data from the a database

♣ INSERT - insert data into a table

♣ UPDATE - updates existing data within a table

♣ DELETE - deletes all or specific (+WHERE) records from a table

DCL (Data Control Language) statements:

♣ GRANT - gives user's access privileges to database

♣ REVOKE - withdraw access privileges given with the GRANT command

Interview Questions

1. What is a Database (dB)?

2. What is a Relational database? Advantages?

3. What is a Schema? What are the schema commands?

4. What is a Table? Can we have an empty table?

5. What is Field? What ‘part’ of the table?

6. What is Record? What ‘part’ of the table?

7. What is Query? 8. What is Report? How much information can you get in one report?

9. What is DBMS? Name a few popular databases.

10. What is a PRIMARY KEY (PK) / FOREIGN KEY (FK)?

Interview Questions

1. What is a database NORMALIZATION?

2. What is SQL stands for? What does it do?

3. Name a few Schema commands; Describe.

4. How to add a record to the table?

5. How to change one field in the table? How to change table column?

7. What is the difference between DROP, DELETE and TRUNCATE?

1. What is Constraint? Give couple constraints examples?

2. How to read data from dB?

3. What are the minimum requirements for the SELECT statement?

4. Is WHERE required in the SELECT statement? Why?

5. How can I retrieve information for the whole table?

6. How would you read all first names from CUSTOMER table?

7. Is it possible to have duplicate first names in CUSTOMER

table? How would you read only unique names?

8. How would you retrieve all employees 55+ years old, living

in cities, starting with ‘San’, showing the oldest employee first?

9. How would you read all employees who was born in Sep, Oct, and Nov?

Interview Questions

1. How would you choose all employees who do NOT have

the address, stored in the system?

2. GROUP BY, HAVING, ORDER BY are required. True or

False? Why?

3. How can I group results by specified criteria?

4. When are we using HAVING option?

5. What is an aggregate function? Examples?

6. How can I count all records within a table?

7. How can I count all the records within the department?

8. How can I find the average salary?

SELECT [ALL | DISTINCT] column1[,column2]

FROM <table1> [,<table2>]

[WHERE "conditions"]

[GROUP BY "column-list"]

[HAVING "conditions]

[ORDER BY "column-list" [ASC | DESC]]

 SELECT… FROM… are the only two required keywords;

 WHERE, GROUP BY, HAVING, ORDER BY are optional, but must be in the

specific order;

 FROM - must contain at least one table from where we are retrieving

information;

 WHERE – optional. If used, specify any condition(s) by using operators:

=, >, <, >=, <=, <> (not equal);

AND (both conditions),

OR (one or another, or both);

IN (specify multiple values);

BETWEEN (select a range of data between two values);

LIKE (search for a specified pattern in a column: % or \_ (one character))

IS NULL / IS NOT NULL (operator for an unknown (NULL) value)

 WHERE – examples:

WHERE (DEPARTMENT\_ID > 50 AND <> 100) OR (DEPARTMENT\_ID = 20 )

WHERE DEPARTMENT\_ID IN (80, 90, 100)

WHERE DEPARTMENT\_ID BETWEEN 80 AND 120

WHERE DEPARTMENT\_ID LIKE ‘LIN%

WHERE DEPARTMENT\_ID IS NOT NULL

 Numeric values should not be enclosed in quotes, but text values should:

(SALARY > 200 and NAME = ‘Smith’ )

 GROUP BY – optional, allow to group the results by some criteria;

SELECT DEPARTMENT\_ID, COUNT(\*)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

 HAVING – optional. WHERE doesn’t work for aggregate functions (avg,

count, sum, max, etc). Use HAVING instead;

SELECT DEPARTMENT\_ID, MAX(SALARY)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

HAVING MAX(SALARY) > 10000

 ORDER BY – optional, allow to order the results in specific order (ASC,

default, sorts in A-Z, DESC sorts in Z-A)

 SELECT commands:

SELECT \* FROM… All columns;

SELECT <column1>, <column2> Data in specified columns (1, 2, etc);

SELECT DISTINCT <column1> Unique values in Column1;

SELECT ROWNUM FROM… Number of each row;

Agregate functions:

Any Datatype:

SELECT COUNT() FROM… Total count of rows;

SELECT MAX() FROM… Largest value;

SELECT MIN() FROM… Smallest value;

SELECT FIRST() FROM… First value;

SELECT LAST() FROM… Last value;

Numeric Datatype Only:

SELECT SUM() FROM… Sum value;

SELECT AVG() FROM… Average value;

 Dual table is a temporary table with one dummy column and single row

SELECT SYSDATE FROM DUAL; System Date

SELECT USER FROM DUAL; System User

 VIEW is a virtual table which is created by executing a query. It is an

abbreviation of SELECT statement (tableA\_v)

 MATERIALIZED VIEW – contain physical data. For big tables, frequently use

only some data. Helps execute queries faster;

 TRIGGER is a piece of SQL code that is activated when a certain event

happens. A common usage is when a new record is added to a database, this

triggers is invoked;

 INDEX is an optional structure, associated with tables, created to improve

query performance.

 JOIN - Get information from related tables

- Inner join (default) displays only rows that matched in both joined tables.

- Outer join includes rows even if they do not have related rows in joined table

 Left outer join

 Right outer join

 Full outer join

- Cross Join includes one row for each possible pairing of rows from two tables

Inner JOIN most common (default) – same as SELECT with WHERE clause:

SELECT \*

FROM EMPLOYEES

[INNER] JOIN DEPARTMENTS

ON EMPLOYEES.Department\_ID = DEPARTMENTS. Department\_ID

 UNION- Combine results of 2 or more select statements (distinct only results)

UNION ALL – return all (including duplicates) data back

SELECT column\_name(s) FROM table\_name1

UNION (ALL)

SELECT column\_name(s) FROM table\_name2

Select must have: same columns number, column order, selected columns of the same

types;

 UNION vs. JOIN:

UNION combines the results of two or more queries into a single result set that

includes all the rows that belong to all queries in the union.

JOINs retrieve data from two or more tables based on logical relationships between the tables. Joins indicate how SQL should use data from one table to select the rows in another table

How did you use SQL in your project?

Q. Your experience with SQL

Q. From your previous experience that you had with SQL, can you

scale from 1-10?

Q. What does SQL stand for?

A. Structured Query Language

Q. What is Normalization?

A. The process of table design is called normalization to minimize data

dependency and redundancy.

Describe how NULLs work in SQL?

NULL is how SQL handles missing (UNDEFINED) values. IS NULL or

IS NOT NULL are arithmetic commands. It will return a NULL.

Q. List all the possible values that can be stored in a BOOLEAN data

field.

A. There are only two values that can be stored in a BOOLEAN data

field: -1(true) and 0(false).

Q. What is the highest value that can be stored in a BYTE data field?

A. The highest value that can be stored in a BYTE field is 255. or from

-128 to 127.

Q. How user defined data types and when you should go for them?A. Define data types when creating or altering table. Then you need to

specify column and datatype. For example:

CREATE TABLE table1 (ID int, Last\_Name varchar(255))

ALTER TABLE table1 ADD column1 varchar(255)

ALTER TABLE table1 MODIFY column1 varchar(255)

What is a primary key?

Primary key is a unique column(s) in the table

Q. What is the main role of a primary key in a table?

A. The main role of a primary key in a data table is to maintain the internal

integrity of a data table.

Q. What are foreign keys?

A. Foreign key is a field that links one table to another table’s primary or

foreign key. Usually it’s not unique

Q. Can a table have more than one foreign key defined?

A. Table can have any number of foreign keys defined. It can have only one

primary key defined.

Q. What are constraints? Explain different types of constraints.

A. CONSTRAINT restricts the values in dB)

- Not Null constraint - dB value can’t be null

- Unique constraint - multiple rows can’t have same value in the same column

- Primary Key constraint - combines a NOT NULL and Unique constraints

- Foreign Key constraint

Q. What is DML, DDL, DCL?

DML and DDL are subsets of SQL.

DDL (Data Definition Language) statements are defining dB schema. DDL

commands are CREATE, ALTER, DROP, RENAME, TRUNCATE, for TABLES,

INDEXES, VIEWS, etc.

DML (Data Manipulation Language) statements are managing data within

schema. DML commands are SELECT, INSERT, UPDATE, DELETE

DCL (Data Control Language) statements: GRANT, REVOKE

Q. Which of the following statements are Data Manipulation Language

commands (DML)?

A.INSERT DML (Data Manipulation Language)

B.UPDATE DML (Data Manipulation Language)

C.TRUNCATE DDL (Data Definition Language)

D.CREATE DDL (Data Definition Language)

E.GRANT DCL (Data Control Language)A. A and B. The INSERT and UPDATE statements are Data Manipulation

Language (DML) commands.

Q. Difference between TRUNCATE, DELETE and DROP commands?

A. DELETE is used to remove some or all (no WHERE clause) rows from a

table;

TRUNCATE removes ALL rows from a table and cannot be rolled back;

DROP removes a table from the database, the tables’ rows, indexes and

privileges;

Q. SQL Update table query. How could you update a cell in SQL table?

A. UPDATE table1

SET column1 = 'Smith', column2 = 200

WHERE column3 = 'ABC‘

Q. How to solve problem with UPDATE one cell in SQL and not change all the

parameters in a column?

A. Add WHERE clause to specify your conditions and set the appropriate value:

UPDATE table1

SET column1 = 'Smith‘ <---- new value

WHERE column2 = 'ABC‘ <---- specified conditions

That would not change column parameters. ALTER can change the column

parameters (length, datatype, add and delete column, etc);

Q. Is the WHERE clause must always appear before the GROUP BY clause

in SQL SELECT ?

A. Yes. Only the SELECT and FROM clause are mandatory. The proper order for

SQL SELECT clauses is:

SELECT ….

FROM ….

WHERE ….

GROUP BY….

HAVING ….

ORDER BY….

Q. Is the WHERE clause must ALWAYS appear in SQL SELECT ?

A. No. Only the SELECT and FROM clause are mandatory. WHERE clause is used

to specify the certain condition, not select the whole table.

Q. When do you use WHERE clause and when do you use HAVING clause?

A. WHERE doesn’t work for aggregate functions (avg,count, sum, max, etc). We

use HAVING clause instead;

SELECT DEPARTMENT\_ID, max(SALARY)

FROM EMPLOYEES

GROUP BY DEPARTMENT\_ID

HAVING MAX(SALARY) > 10000

Q. What is difference between Rename and Alias?

A. When you RENAME table or column (ALTER) you actually changing the

name, when you use alias, you are just giving a table or column a

“nickname”

Q. Describe SQL comments.

A. SQL comments are introduced by two consecutive hyphens (--) and ended

by the end of the line.

Q. What does COMMIT do? ROLL BACK?

COMMIT save all changes made by DML statements to the dB, ROLL BACK

will undo all changes

Q. What is the system function to get the current user’s user id?

A. SELECT USER FROM DUAL

Q. How do you add record to a table?

A. INSERT into table\_name VALUES (‘ALEX’, 33 , ‘M’)

Q. How do you add a column to a table?

A. ALTER TABLE Department ADD (AGE, NUMBER)

Q. How do you change value of the field?

A. UPDATE table\_name

SET field1 = 200,

field2 = ‘ABD’,

field3 = to\_date ('2006-03-04 09:29', 'yyyy-mm-dd')

WHERE field = ‘CD’ (CONDITION)

Q. How do you find the number of rows in a Table?

A. SELECT count(\*) FROM table\_name

Q. How do you select all records from the table?

A. SELECT \* FROM table\_name

Q. Write a SQL SELECT query that returns each city only once from Students

table? Do you need to order this list with an ORDER BY clause?

A. SELECT DISTINCT City FROM Students;

Yes, you need an ORDER BY ASC or DESC, if you need to sort more than

one returned CITY

Q: Your database contains customers and their addresses. What SQL

command you will use to retrieve all addresses within a specific city?

A. SELECT last\_name, first\_name, address

FROM EMPLOYEES

WHERE EMPLOYEES.CITY = ‘San Francisco’

Q. Write SQL SELECT query that returns the first and last name of each

instructor, Salary, and gives each of them a number.

A. SELECT FirstName, LastName, Salary, ROWNUM

FROM Instructors;

Q. Can one select a random collection of rows from a table?

A. Yes. Using SAMPLE clause.

Example:

SELECT \* FROM EMPLOYEES SAMPLE(10);

10% of rows selected randomly will be returned

Q: There is a % sign in one field of a column. What will be the query to find it?

A. SELECT column1

FROM table\_name

WHERE column1 LIKE '%[%]%'

Q. Choose all ‘5’ from the field.

A. SELECT column1

FROM table\_name

WHERE column1 LIKE '%5%'

Q. You issue the following query:

SELECT First\_Name FROM EMPLOYEES

WHERE First\_Name LIKE '\_A%'

Which names would be returned by this query? Choose all that apply.

A. Allen

B. CLARK

C. JACKSON

D. David

A. A and C. '\_A% is one missing character (\_) and any character(s) after A (%)

Q. What is Trigger?

A. TRIGGER defines a set of actions that are performed in response to an

insert, update, or delete operation on a specified table. When such an SQL

operation is executed, in this case the trigger has been activated.

Q**. What is Index? (Advanced: What are the types of indexes, etc.?)**

A. INDEX is an optional structure, associated with tables, created to improve

query performance.

**Q. What is a difference between View and Materialized View?**

A. VIEW is a virtual table which is created by executing a query. It is an

abbreviation of SELECT statement (tableA\_v)

MATERIALIZED VIEW – contain physical data. For big tables,

frequently use only some data. Helps execute queries faster;

**Q. What is a join?**

A. Join is a process of retrieve pieces of data from different sets (tables) and

returns them to the user or program as one “joined” collection of data.

**Q. How do you extract information from two tables using SQL?**

A. You can use SELECT…WHERE table1.id = table2.id or use JOIN

Q. **What is INNER JOIN?**

A. Inner Join displays only rows that matched in both joined tables. It’s a default

and most commonly used join. It is a different syntax for select \* from table1,

table2 where table1.id=table.id

SELECT \*

FROM EMPLOYEES

[INNER] JOIN DEPARTMENTS

ON EMPLOYEES.Department\_ID = DEPARTMENTS. Department\_ID

**Q. What is OUTER JOIN?**

A. Outer join includes rows even if they don’t have related rows in joined table.

3 types of outer join are: Left outer join, Right outer join, Full outer join

**Q. What is CROSS\_JOIN?**

A. Cross Join includes one row for each possible pairing of rows from two tables

**Q. What is a self join?** Explain it with an example.

A. Self-join is a query in which a table is joined to itself. Example: Employees table which

contains rows for normal employees as well as managers. So, to find out the managers

of all the employees, you need a self join.

SELECT e.Last\_Name, m.Last\_Name

FROM EMPLOYEES e, EMPLOYEES m

WHERE e.ID = m.MGRID

Q. Write query to get unique zipcodes from a table which has state, country and zipcodes.

A. SELECT DISTINCT ZIPCODE FROM ADDRESS

Q. Write query to get the unique zip codes from 2 tables.

Table "A" has SSN, State, Zipcodes, and Table "B" has Zipcode, City,

A. SELECT DISTINCT ZIPCODE FROM TABLEA

UNION

SELECT DISTINCT ZIPCODE FROM TABLEB

Q. Table with state and zipcode columns. Write a query to get unique zipcodes from each

state?

SELECT DISTINCT state, zipcode

FROM LOCATION

Q: There is a % sign in one field of a column. What will be the query to find it?

A. SELECT column1 FROM table\_name

WHERE column1 LIKE '%[%]%'

Q. Which SQL statement would you use to extract data from a database?

A. SELECT

Q. How can you change “12-03-1967” to “12-23-1967” in DOB (date of birth)

column in Employees table?

A. UPDATE Employees SET DOB = ’12-23-1967’ WHERE DOB = ’12-03-1967’

The whole column DOB will be updated without WHERE clause

Q. Which statement would you use to delete data from a database?

A. DELETE FROM City

WHERE city = ‘San Francisco’

Q. Please select all records from table “Employees” where value of the column

“LastName” contains “ba”;

A. SELECT \* FROM EMPLOYEES WHERE LastName LIKE '%ba%'

Q. How would you sort the results?

A. SELECT \* FROM EMPLOYEES

ORDER BY LastName ASC (default a-z), DESC (z-a)

**Difference between Where and Having ?**

Where 🡪 filtering the result from select query

Having 🡪 filtering the result after performing aggregate functions

-------

**Select only duplicaterows**

Select columnname, count(\*)

from tablename

where columnname = value

Group by columnname

Having count(\*) > 1

**Select non duplicaterows**

Select columnname, count(\*)

from tablename

where columnname = value

Group by columnname

Having count(\*) = 1

Or use Distinct for displaying only distinct row values including which have duplicates

**Can foreign keys have null values?**

Yes. If value is available, it should be a value from the base table else error will be displayed

**What is the use of On Delete Cascade 🡪 when creating a child table with foreign key reference**

When parent data is deleted, corresponding data is deleted automatically from child table

**Create table with foreign key**

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

ETL testing – approach

1. Duplicate values
2. Row count
3. Constraint check
4. Data integrity
5. Data truncation
6. Sample test with some rows
7. Minus query for data from staging 🡪 datawarehouse ETL process

**DDL , DML, DCL, TQL , DQL**

DQL – Data Query Language 🡪 Select query 🡪 fetching data

DDL 🡪 Data definition Language 🡪 Create / Alter (operations on structure)

DML 🡪 Data manipulation Language 🡪 insert / update / delete (operations on data)

DCL 🡪 Data Control Language 🡪 grant and revoke (permission/ access)

TQL 🡪 Transaction Query Language 🡪 Rollback(revert the changes in the db after dml statements not after commit), commit (Save the changes to the database after dml statements), savepoint ( will hold the changes up to that point)

**Active Transformation**

Source no of records is not same as target no of records (because of filtration)

**Passive Transformation**

Source no of records= target no of records (transformation does not affect the no of records)

Types of filterations

**SQ – Source Qualifier** (direct inner join from source and getting the result into a target table)

**Filter transformation** – filtering the records based on transformation conditions.

(Check all details in workflow monitor / manager in informatica power client tool)

Star Schema and Snowflake Schema

|  |  |
| --- | --- |
| **Star Schema** | **Snowflake Schema** |
| Hierarchies for the dimensions are stored in the dimensional table. | Hierarchies are divided into separate tables. |
| It contains a fact table surrounded by dimension tables. | One fact table surrounded by dimension table which are in turn surrounded by dimension table |
| In a star schema, only single join creates the relationship between the fact table and any dimension tables. | A snowflake schema requires many joins to fetch the data. |
| Simple DB Design. | Very Complex DB Design. |
| Denormalized Data structure and query also run faster. | Normalized Data Structure. |
| High level of Data redundancy | Very low-level data redundancy |
| Single Dimension table contains aggregated data. | Data Split into different Dimension Tables. |
| Cube processing is faster. | Cube processing might be slow because of the complex join. |
| Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions. | The Snowflake schema is represented by centralized fact table which unlikely connected with multiple dimensions. |

**A Galaxy Schema** contains two fact table that share dimension tables between them. It is also called Fact Constellation Schema. The schema is viewed as a collection of stars hence the name Galaxy Schema.

Fact table and Dimension table

| **Parameters** | **Fact Table** | **Dimension Table** |
| --- | --- | --- |
| **Definition** | Measurements, metrics or facts about a business process. | Companion table to the fact table contains descriptive attributes to be used as query constraining. |
| **Characteristic** | Located at the center of a star or snowflake schema and surrounded by dimensions. | Connected to the fact table and located at the edges of the star or snowflake schema |
| **Design** | Defined by their grain or its most atomic level. | Should be wordy, descriptive, complete, and quality assured. |
| **Task** | Fact table is a measurable event for which dimension table data is collected and is used for analysis and reporting. | Collection of reference information about a business. |
| **Type of Data** | Facts tables could contain information like sales against a set of dimensions like Product and Date. | Evert dimension table contains attributes which describe the details of the dimension. E.g., Product dimensions can contain Product ID, Product Category, etc. |
| **Key** | Primary Key in fact table is mapped as foreign keys to Dimensions. | Dimension table has a primary key columns that uniquely identifies each dimension. |
| **Storage** | Helps to store report labels and filter domain values in dimension tables. | Load detailed atomic data into dimensional structures. |
| **Hierarchy** | Does not contain Hierarchy | Contains Hierarchies. For example Location could contain, country, pin code, state, city, etc. |

Minus query – Sample (Except in SQL Server and Minus in Oracle)

SELECT expression1, expression2, ... expression\_n

FROM tables

[WHERE conditions]

**EXCEPT 🡪 MINUS in oracle**

SELECT expression1, expression2, ... expression\_n

FROM tables

[WHERE conditions];

There must be same number of expressions in both SELECT statements.

The corresponding expressions must have the same data type in the SELECT statements. For example: expression1 must be the same data type in both the first and second SELECT statement.

This EXCEPT / MINUS operator example returns all values that are in the left table and not in the right table. What this means is that if a value existed in the left table and also existed in the right table, the value would not appear in the EXCEPT / MINUS query results.

Router Transformation

For different conditions like using case statement instead of if condition

It is a kind of passive transformation where data is not lost or truncated. It is routed to default.

|  |  |
| --- | --- |
| **Router** | **Filter** |
| Router transformation provides us the facility to capture the rows of data that do not meet any of the conditions to a default output group. | A Filter transformation tests data for one condition and drops the rows of data that do not meet the condition. |
| Router transformation is a single input and multi output group transformation. | Filter is single input and single output group transformation. |
| In a router transformation, you can specify more than one filter condition. | Filter transformation, you can specify only one filter condition. |
| The router transformation does not block input rows and those records that failed the filter condition will be passed to the default group | In a filter transformation there is chance that records get blocked |
| Router transformation acts like IIF condition in informatica or CASE.. WHEN in database. | Filter transformation works as WHERE clause of SQL . |

Sorter transformation 🡪 ordering the data 🡪 use order by

This is active transformation ( removes the duplication)

SCD – Slowly changing dimension

Dimension table changing according to certain conditions or rules

* Type 0 – Fixed Dimension
  + No changes allowed, dimension never changes
* Type 1 – No History
  + Update record directly, there is no record of historical values, only current state
* Type 2 – Row Versioning
  + Track changes as version records with current flag & active dates and other metadata
* Type 3 – Previous Value column
  + Track change to a specific attribute, add a column to show the previous value, which is updated as further changes occur
* Type 4 – History Table
  + Show current value in dimension table but track all changes in separate table
* Type 6 – Hybrid SCD
  + Utilise techniques from SCD Types 1, 2 and 3 to track change