SQL

Aug 02 & Aug 03,2025

How is SQL pronounced?

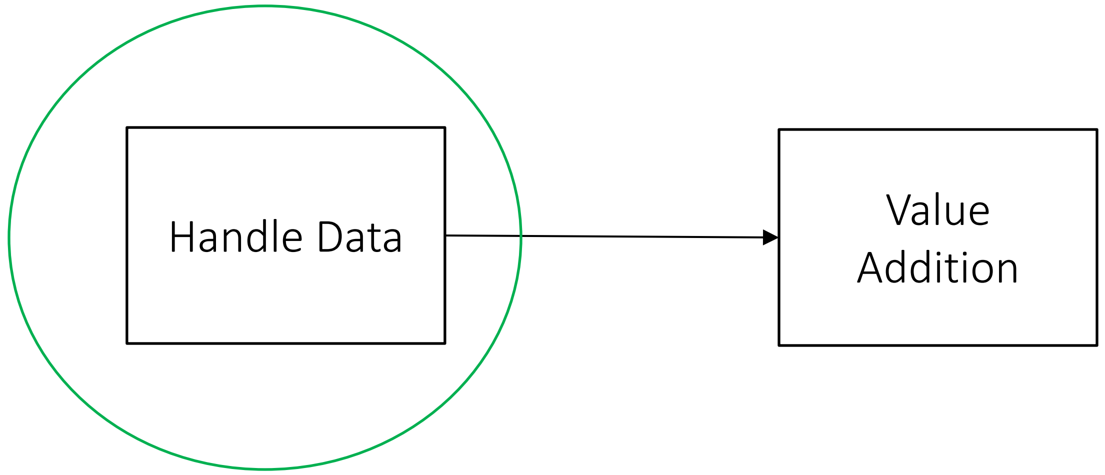
Sequel

S.Q.L

What is SQL?

Structured Query Language

Why SQL?



Day 1 Agenda

Information System

Database Management Systems

Overview Of SQL

Data Definition Language

Data Manipulation Language

Constraint

Retrieval-The Basics

Information Systems

Organizations need formal systems that will allow them to produce the required information, in the right format, at the right time.

Such systems are called information systems.

An information system is a simplified reflection (a *model*) of the real world within

the organization.

Information systems don’t necessarily need to be automated—the data might reside in physical storage mechanisms. This data can be converted into the desired information using certain procedures or actions.

Information Systems

In general, there are two main reasons to automate information systems:

Complexity

Volume

Database Management Systems

Consider a company in the supermarket business,

What information do you collect?

Database Technology

If an organization decides to automate an information system because of complexity or volume (or both), it typically will need to use some database technology.

Oracle

Microsoft SQL

MySQL

IBM DB2

Hive

Teradata

Granularity Of Data

Every dataset has dimensions. Lowest level of data which answers the three WH questions:

1. WHEN?
2. WHERE?
3. WHAT?

Query Languages

- Key component of DBMS

- SQL-de facto market standard for many years

RDBMS

- Relational Database Management System (RDBMS) was laid out in 1970 by Ted Codd

- He derived his revolutionary ideas from classical components of mathematics: set theory, relational calculus, and relational algebra.

RDBMS

Basic relational data structures are as follows:

- A database, which is a set of tables
- A table, which is a set of rows

Overview of SQL

- Data Definition Language, or DDL:

  Used to define the structure of the database such as creating, altering or deleting tables

- Data Manipulation Language, or DML:

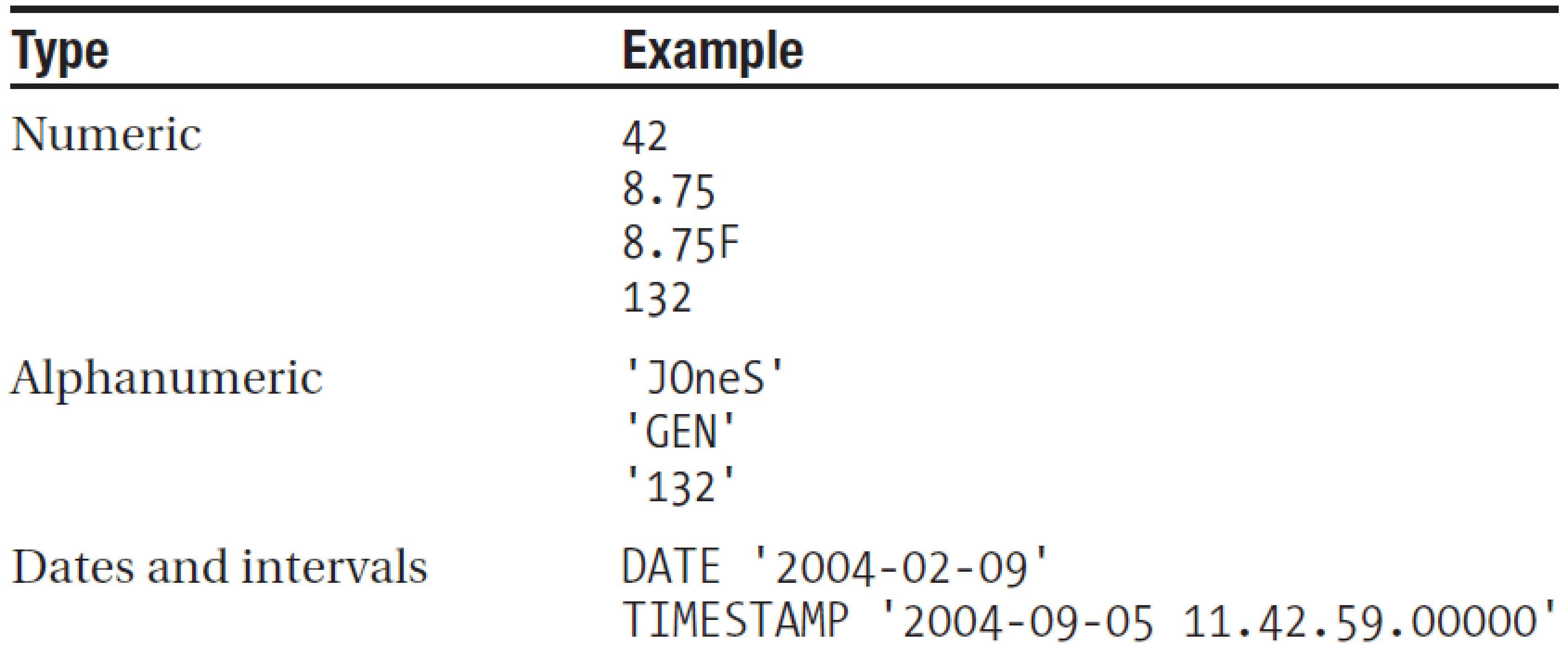
  Used to manipulate data within the database

- Retrieval

Overview of SQL-Constants(Literals)

- A *constant* (or *literal*) is something with a fixed value.
- Numbers (numeric constants)
- Text (alphanumeric constants),alphanumeric constants are also referred to as *strings*.
- In the SQL language, alphanumeric constants (strings) must be placed between *single quotation marks (*quotes).

Overview of SQL-Constants(Literals)



Type	Example
Numeric	42 8.75 8.75F 132
Alphanumeric	'JOneS' 'GEN' '132'
Dates and intervals	DATE '2004-02-09' TIMESTAMP '2004-09-05 11.42.59.00000'

Datatypes

Datatype	Description
Integer	Whole Numbers
Varchar(n)	Variable-length string; maximum *n* characters
Double	Floating point numbers
Date	Date

Overview of SQL-Operators

The SQL operators are divided in various categories, where the differentiating factor is the operand datatype:

- Arithmetic operators
- Comparison operators
- Logical operators

Overview of SQL-Operators

Arithmetic

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division

Comparison

Operator	Description
<	Less than
>	Greater than
=	Equal to
<=	Less than or equal to
>=	Greater than or equal to
<> or !=	Not equal to

Logical

Operator	Description
AND	Logical AND
OR	Logical OR (the *inclusive* OR)
NOT	Logical negation

Overview of SQL-Comments

You can add comments in two ways:

between /* and */ or

after two consecutive minus signs.

Overview of SQL-Reserved Words

Just like any other language, SQL has a list of reserved words.

These are some examples of SQL reserved words: AND, CREATE, DROP, FROM, GRANT, HAVING, INDEX, INSERT, MODIFY, NOT, NULL, NUMBER, OR, ORDER, RENAME, REVOKE, SELECT, SYNONYM, SYSDATE,

TABLE, UPDATE, USER, VALUES, VIEW, and WHERE.

Data Definition Language

CREATE DATABASE

```
CREATE DATABASE DATABASENAME
```

CREATE TABLE

```
CREATE TABLE DATABASE. TABLENAME(COLUMNNAME1 DATATYPE, COLUMNNAME2
DATATYPE,PRIMARY KEY(COLUMNNAME1))
```

DESCRIBE

```
DESC DATABASE.TABLENAME
```

Data Manipulation Language

INSERT: To add rows to a table

INSERT INTO DATABASE. TABLENAME VALUES(VALUE1,VALUE2..,VALUEN)

Data Definition Language

TRUNCATE-Allows you to delete all the rows in a table by maintaining the table structure.

```
TRUNCATE TABLE DATABASE.TABLENAME
```

Change column Datatype

```
ALTER TABLE DATABASE.TABLENAME ADD(COLUMNNAME1 DATATYPE)
```

Data Manipulation Language

UPDATE: To change column values of existing rows

UPDATE DATABASE. TABLENAME SET COLUMNNAME=VALUE

UPDATE DATABASE. TABLENAME SET COLUMNNAME=VALUE WHERE COLUMNNAME=VALUE

Data Definition Language

- MODIFY COLUMN DATATYPE

ALTER is used to Change the table structure

ALTER TABLE DATABASE.TABLENAME MODIFY COLUMNNAME1 DATATYPE

- RENAME COLUMN NAME

ALTER TABLE DATABASE.TABLENAME RENAME COLUMNNAME TO NEWCOLUMNNAME

- RENAME TABLE NAME

RENAME TABLE DATABASE.TABLENAME TO DATABASE.NEWTABLENAME ;

DML & DDL

- **DELETE:** To remove rows from a table

DELETE FROM DATABASE.TABLENAME WHERE COLUMNNAME=VALUE

- **DROP COLUMN:** Delete the column

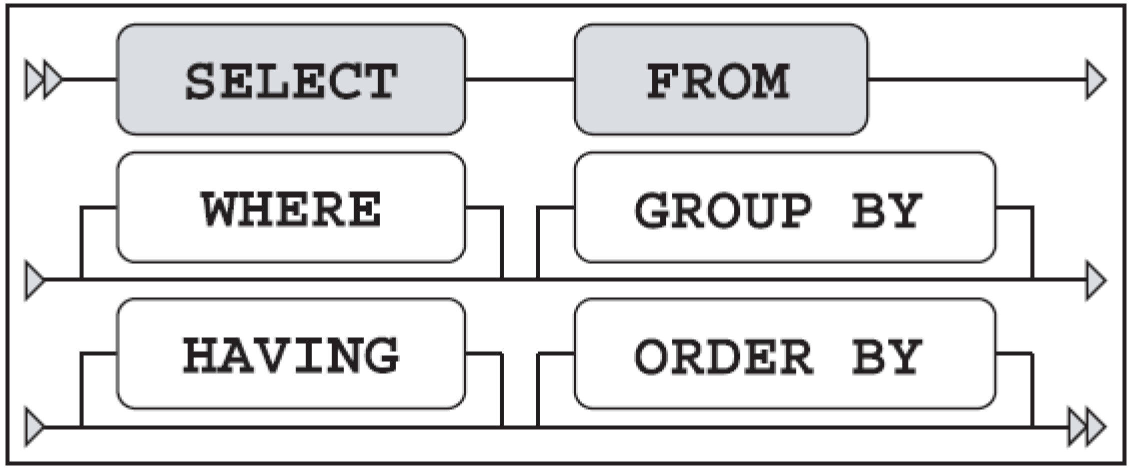
ALTER TABLE DATABASE.TABLENAME DROP COLUMN COLUMNAME

- **DROP TABLE:** Delete the table

DROP TABLE DATABASE.TABLENAME

Retrieval: The Basics

The six main clauses of the SELECT command



6 Main Clauses

SELECT-Which columns do you want to see the result?

FROM-Which table(s) is (are) needed for retrieval?

WHERE- What is the condition to filter the rows?

ORDER BY- In which order do you want to see the resulting rows?

GROUP BY- How should the rows be grouped/aggregated?

HAVING- What is the condition to filter the aggregated groups?

Retrieval: The Basics

*****- Retrieves all columns from a table without needing to specify each column name individually

Alias-Temporary name given to a table or column in a query

DISTINCT- Unique values in a column

BETWEEN-Between x and y, including boundaries

IN -Equals a value from a set (or subquery)

LIKE-Check alphanumeric values using wildcards

Null Values

Subquery- A query within another query

LIMIT-used to specify the maximum number of rows that a query should return

CASE WHEN

```
CASE **expr**
  WHEN **expr_to_match** THEN result
  [ ... ]
  [ ELSE else_result ]
  END
```

Compares expr to expr_to_match of each successive WHEN clause and returns the first result where this comparison evaluates to TRUE. The remaining WHEN clauses and else_result aren't evaluated.

Day 2 Agenda

Retrieval

Function

Join Operations

CTE

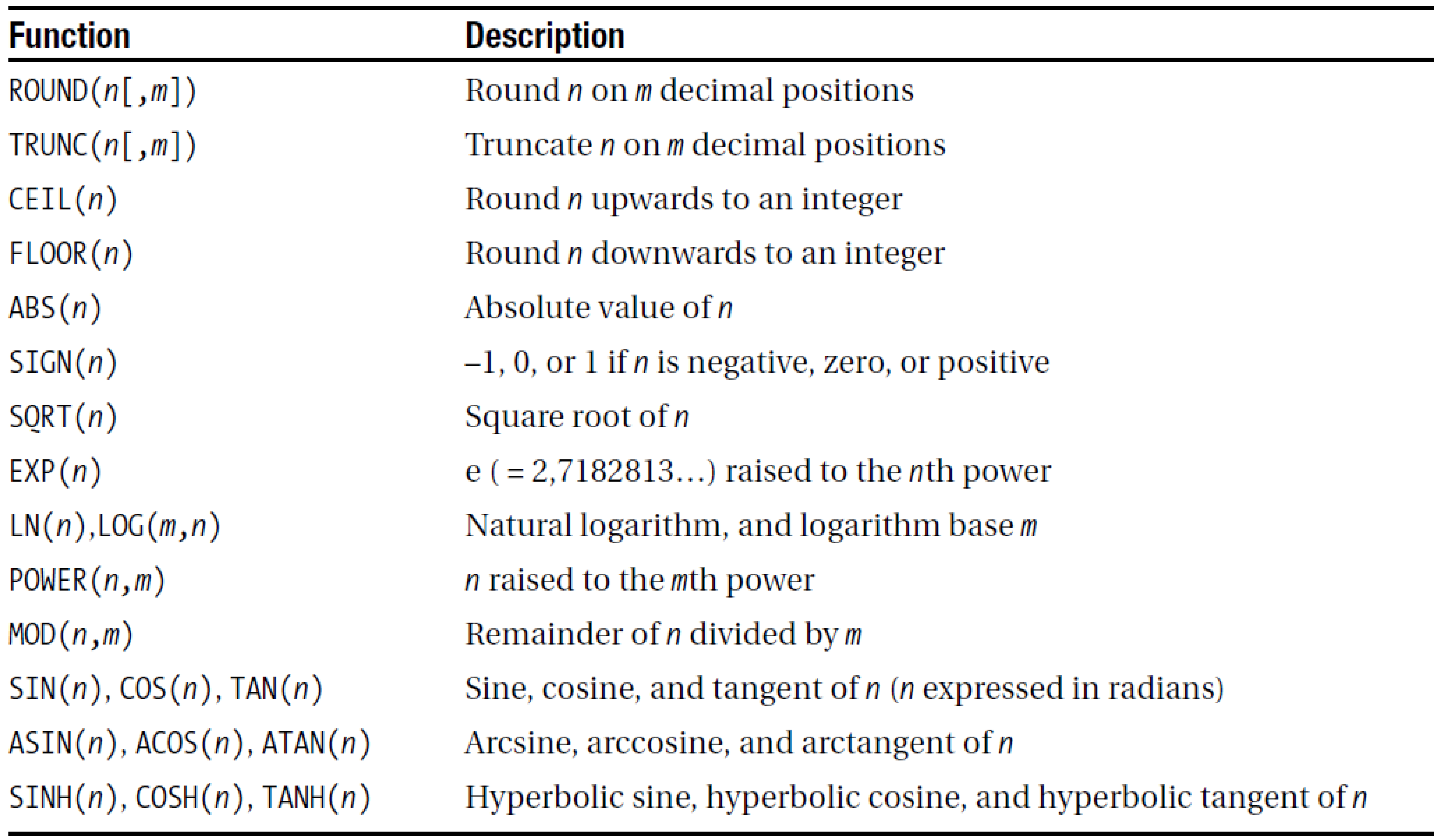
View

Analytical Functions

Retrieval: Functions

Function Type	Applicable To
Arithmetic	Numeric data
Text	Alphanumeric data
Date	Date/time related data

Retrieval: Arithmetic Functions



Function	Description
`ROUND(`*n*`[,`*m*`])`	Round *n* on *m* decimal positions
`TRUNC(`*n*`[,`*m*`])`	Truncate *n* on *m* decimal positions
`CEIL(`*n*`)`	Round *n* upwards to an integer
`FLOOR(`*n*`)`	Round *n* downwards to an integer
`ABS(`*n*`)`	Absolute value of *n*
`SIGN(`*n*`)`	–1, 0, or 1 if *n* is negative, zero, or positive
`SQRT(`*n*`)`	Square root of *n*
`EXP(`*n*`)`	e (= 2,7182813…) raised to the *n*th power
`LN(`*n*`),LOG(`*m*`,`*n*`)`	Natural logarithm, and logarithm base *m*
`POWER(`*n*`,`*m*`)`	*n* raised to the *m*th power
`MOD(`*n*`,`*m*`)`	Remainder of *n* divided by *m*
`SIN(`*n*`), COS(`*n*`), TAN(`*n*`)`	Sine, cosine, and tangent of *n* (*n* expressed in radians)
`ASIN(`*n*`), ACOS(`*n*`), ATAN(`*n*`)`	Arcsine, arccosine, and arctangent of *n*
`SINH(`*n*`), COSH(`*n*`), TANH(`*n*`)`	Hyperbolic sine, hyperbolic cosine, and hyperbolic tangent of *n*

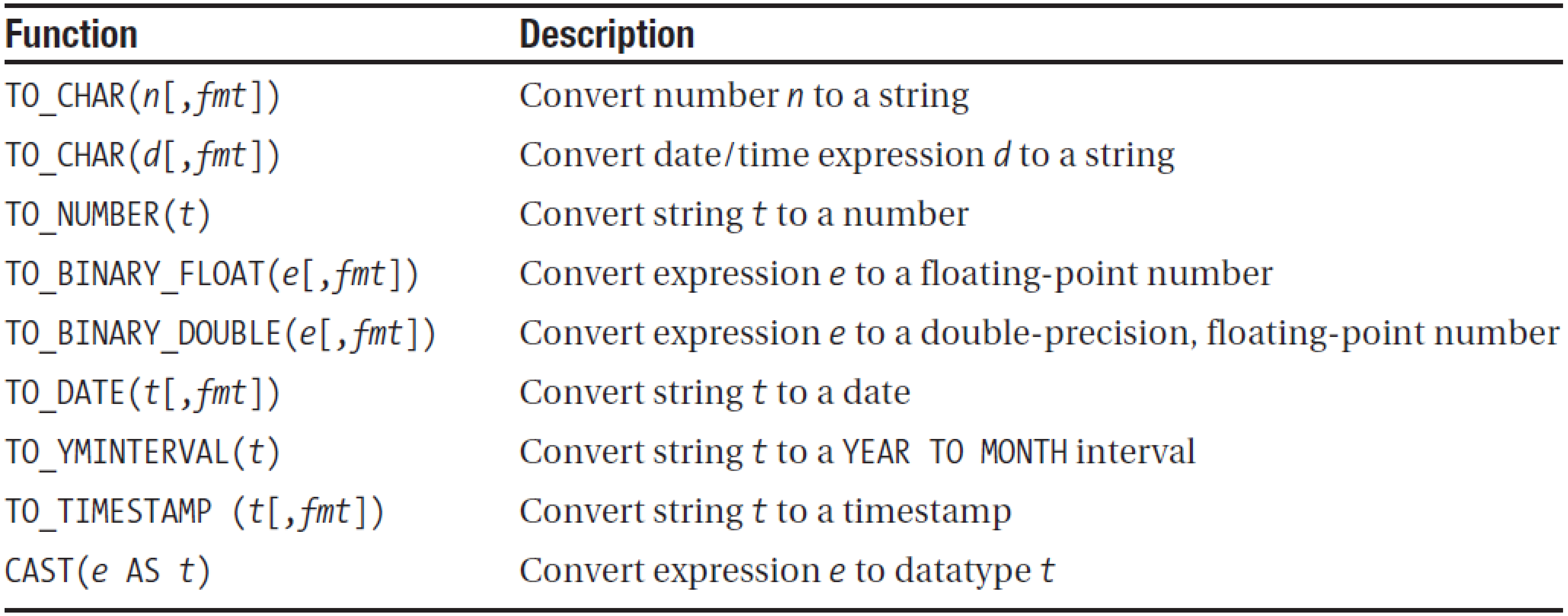
Retrieval: Text Functions

Function	**Description**
LENGTH(*t*)	Length (expressed in characters) of *t*
ASCII(*t*)	ASCII value of first character of *t*
CHR(*n*)	Character with ASCII value *n*
UPPER(*t*), LOWER(*t*)	*t* in uppercase/lowercase
INITCAP(*t*)	Each word in *t* with initial uppercase; remainder in lowercase
LTRIM(*t*[,*k*])	Remove characters from the left of *t*, until the first character not in *k*
RTRIM(*t*[,*k*])	Remove characters from the right of *t*, after the last character not in *k*
TRIM([[*option*][*c* FROM]]*t*)	Trim character *c* from *t*; *option* = LEADING, TRAILING, or BOTH
LPAD(*t*,*n*[,*k*])	Left-pad *t* with sequence of characters in *k* to length *n*
RPAD(*t*,*n*[,*k*])	Right-pad *t* with *k* to length *n* (the default *k* is a space)
SUBSTR(*t*,*n*[,*m*])	Substring of *t* from position *n*, *m* characters long (the default for *m* is until end)
INSTR(*t*,*k*)	Position of the first occurrence of *k* in *t*
INSTR(*t*,*k*,*n*)	Same as INSTR(*t*,*k*), but starting from position *n* in *t*
INSTR(*t*,*k*,*n*,*m*)	Same as INSTR(*t*,*k*,*n*), but now the *m*th occurrence of *k*
TRANSLATE(*t*,*v*,*w*)	Replace characters from *v* (occurring in *t*) by corresponding character in *w*
REPLACE(*t*,*v*)	Remove each occurrence of *v* from *t*
REPLACE(*t*,*v*,*w*)	Replace each occurrence of *v* in *t* by *w*
CONCAT(*t1*,*t2*)	Concatenate *t1* and *t2* (equivalent to the || operator)

Retrieval: Date Functions

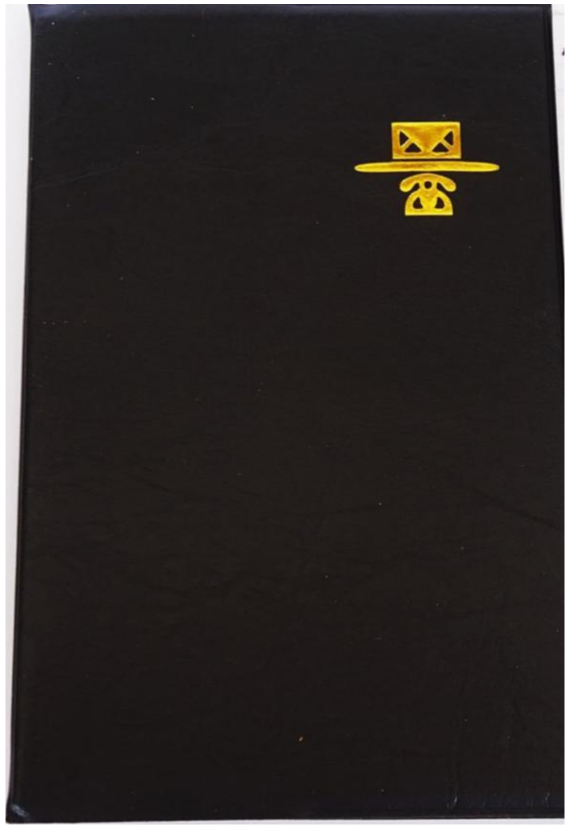
https://dev.mysql.com/doc/refman/8.4/en/date-and-time-functions.html

Retrieval: Conversion Functions



Function	**Description**
TO_CHAR(*n*[,*fmt*])	Convert number *n* to a string
TO_CHAR(*d*[,*fmt*])	Convert date/time expression *d* to a string
TO_NUMBER(*t*)	Convert string *t* to a number
TO_BINARY_FLOAT(*e*[,*fmt*])	Convert expression *e* to a floating-point number
TO_BINARY_DOUBLE(*e*[,*fmt*])	Convert expression *e* to a double-precision, floating-point number
TO_DATE(*t*[,*fmt*])	Convert string *t* to a date
TO_YMINTERVAL(*t*)	Convert string *t* to a YEAR TO MONTH interval
TO_TIMESTAMP (*t*[,*fmt*])	Convert string *t* to a timestamp
CAST(*e* AS *t*)	Convert expression *e* to datatype *t*

Partition



Order Of Execution Of Query

1) FROM and JOIN
2) WHERE
3) GROUP BY
4) HAVING
5) SELECT
6) DISTINCT
7) ORDER BY
8) LIMIT

Order Of Execution Of Query

```
5 SELECT
     CUSTOMERID,
     SUM(QUANTITY) AS TOTAL_QTY
1 FROM
     IMT_OCT23.ONLINE_RETAIL
2 WHERE
     CUSTOMERID IS NOT NULL
3 GROUP BY
     CUSTOMERID
4 HAVING
     TOTAL_QTY>100
6 ORDER BY
     TOTAL_QTY
7 LIMIT 5
```

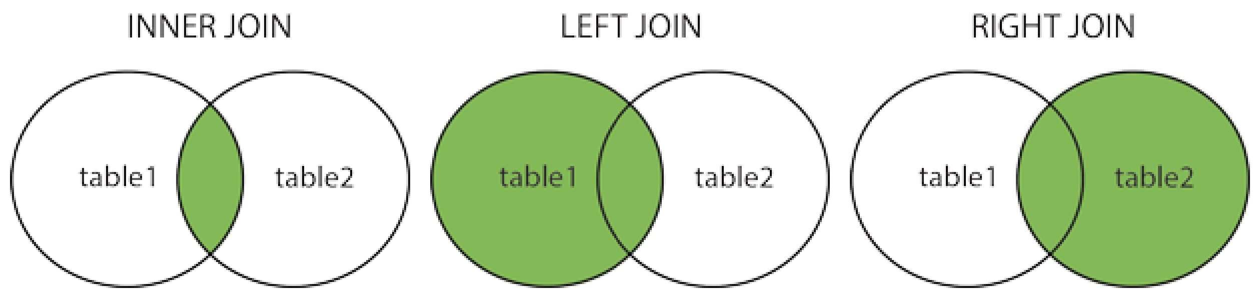
Retrieval: Multiple Tables And Aggregation

Tuple Variables

Tuple variables are referred to as table aliases

Retrieval: Multiple Tables And Aggregation

Joins



Retrieval: Multiple Tables And Aggregation

Inner Join:

It returns only the rows that have matching values in both tables being joined.

If there is no match,the row is excluded from the result.

Left Join:

It returns all the rows from the left table and the matched rows from the right table.

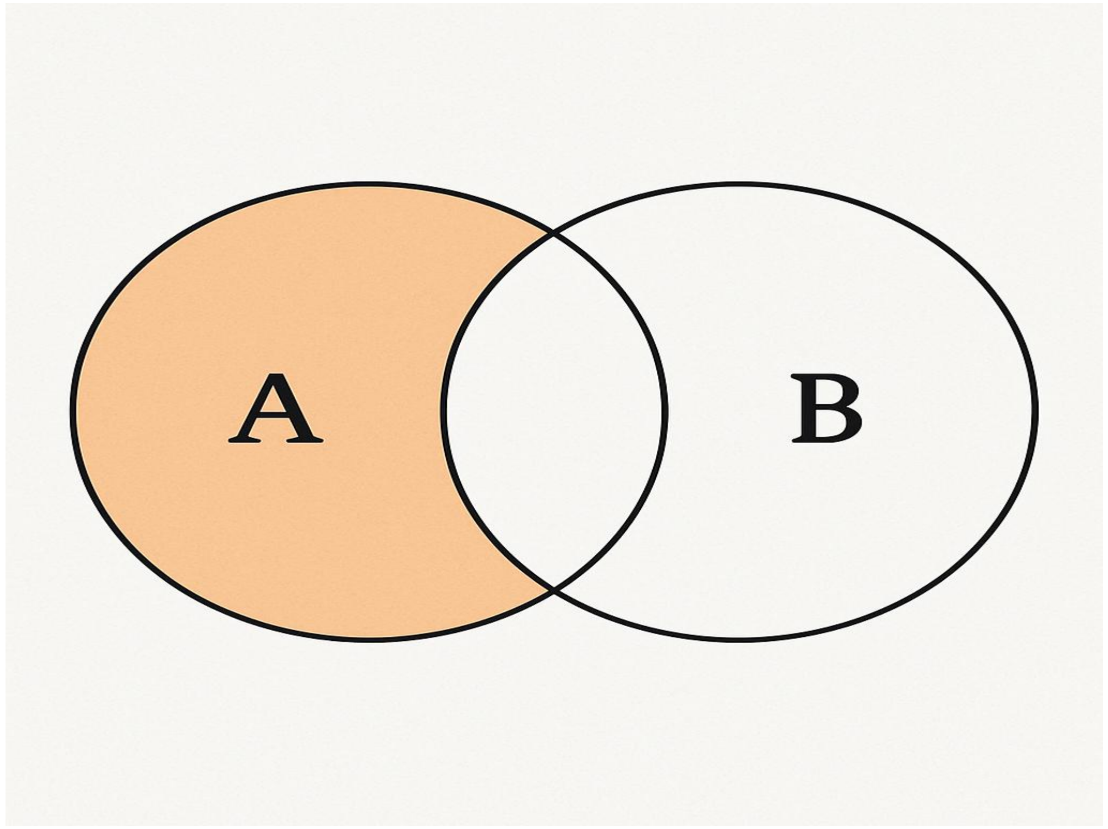
If there is no match, then result is NULL for columns from the right table.

Right Join:

It returns all the rows from the right table and the matched rows from the left table.

If there is no match, then result is NULL for columns from the left table.

Retrieval: Multiple Tables And Aggregation



Common Table Expression(CTE)

A common table expression (CTE) is a named temporary result set that exists within the scope of a single statement and that can be referred to later within that statement, possibly multiple times.

Constraints

Primary Key

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

Union, View

UNION: It is used to combine the results of two or more SELECT queries and **removes duplicate rows.**

UNION ALL: It is used to combine the results of two or more SELECT queries and **keeps duplicate rows.**

VIEW:

Views are virtual tables formed by a query. Views are not updatable.

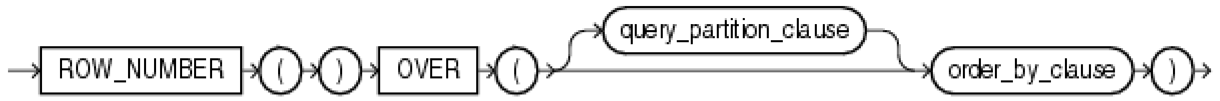
Enhance security by limiting access to specific data and provide an abstraction layer for the underlying table structures.

Analytical Functions

ROW_NUMBER:

It will provide a unique number to each row in resultset.

Syntax

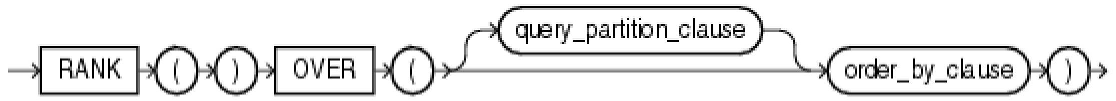


Analytical Functions

RANK:

Rows with equal values for the ranking criteria receive the same rank, then adds the number of tied rows to the tied rank to calculate the next rank.

Therefore, the ranks may not be consecutive numbers.

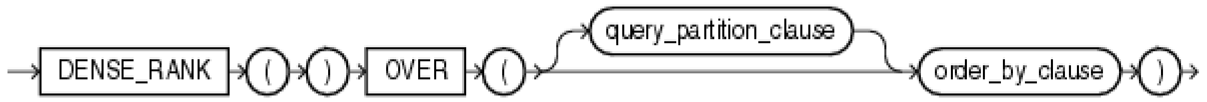


Analytical Functions

DENSE_RANK:

In a normal rank function, we see a gap between the numbers in rows.

DENSE_RANK is a function with no gap.



Best Practices in SQL

Always include Comments for better understanding

Format the code for readability

Check for count after creation of each table

Always use date filter when using select *from