Database Management Systems (CSE221)

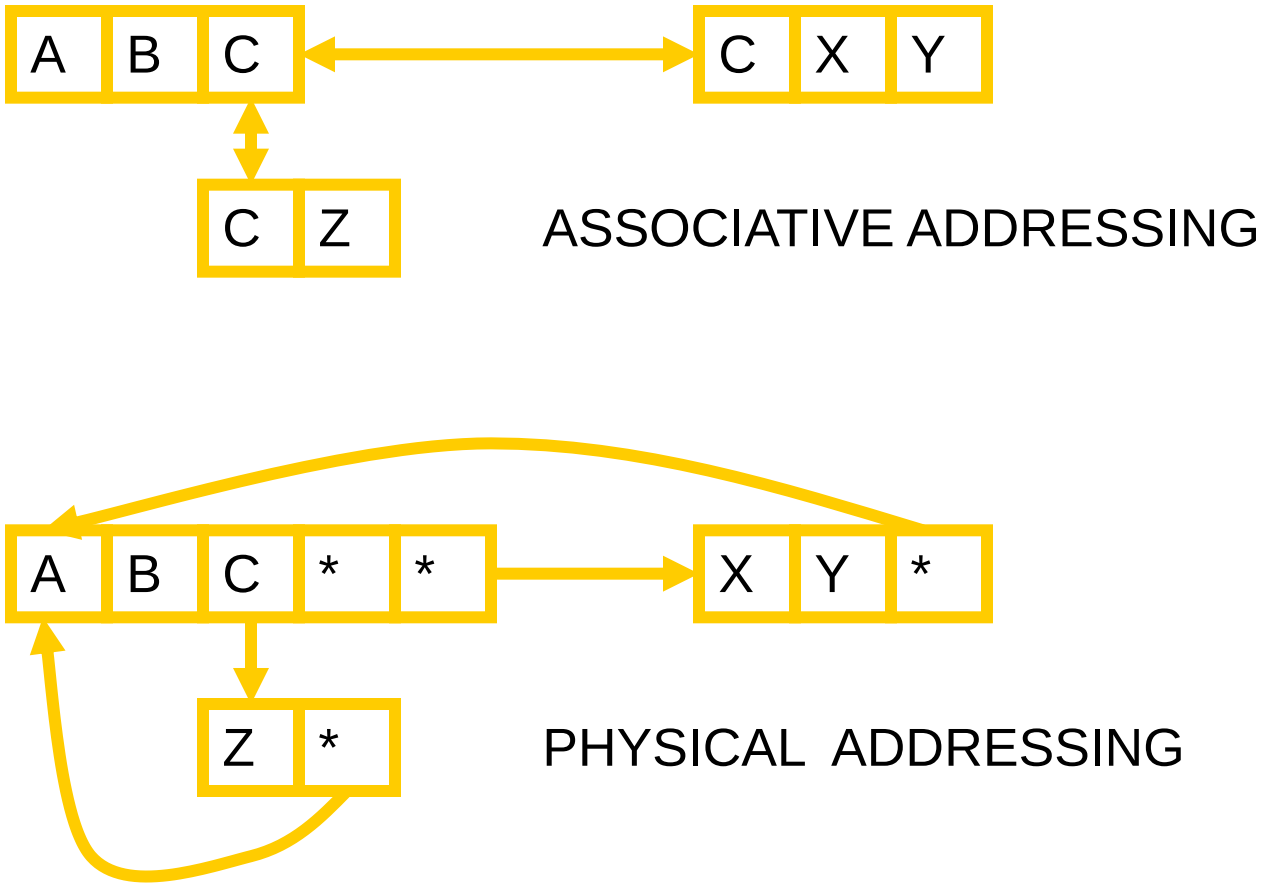
Vikas Bajpai

KEYS

KEYS

- Relational DBMS uses associative addressing.
  - Identify and locate rows by value
  - Physical address is transparent to user

KEYS


A
B
C
C
X
Y
C
Z
ASSOCIATIVE ADDRESSING
A
B
C
*
*
X
Y
*
Z
*
PHYSICAL ADDRESSING

KEYS

- Associative addressing is simpler for the end-user.

- Physical data independence – storage structures and access paths are transparent to user and application programs

KEYS:

- Associative addressing is based on keys – a column, or group of columns, used to identify rows.
- Simple key – a key formed from a single column
- Composite key – a key formed from several columns
- The relational model has five kinds of keys:
  1. Super
  2. Candidate
  3. Primary
  4. Alternate (secondary)
  5. Foreign

KEYS:

- In relational DBMS, a key is not the same as an index!

- Keys identify rows (logical design)
- Indexes locate rows (physical design)

1. Candidate Keys:

- Candidate Key – any (simple or composite) column of a table which is both unique and minimal.
- Uniqueness – no two rows in a table may have same candidate key value at any time.
- Minimality – every column of a composite candidate key must be necessary for uniqueness.

Candidate Keys:

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347

Candidate Keys are individual columns in a table that qualifies for uniqueness of all the rows. In the above table Roll_No and PAN_No are Candidate keys

2. Primary Key

- A Primary key is a key that uniquely identifies a row in each table. It is normally denoted with its first two letters, namely, **PK**
- PK is also a Candidate key.

- A Primary Key can consist of one or more columns as long as the combination of columns is unique.

Primary Key

- Primary Key – a candidate key chosen by the database designer to identify rows of a table in queries
- The primary key is the only guaranteed way to identify rows in queries

```
SQL>UPDATE      COMPENSATION
2. SET          SALARY = 30000
3. WHERE        EMP# = E3;
```

- Primary keys must be unique, minimal, non-null, and preferably time-invariant.

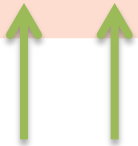
Primary Key

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347

Roll_No uniquely identifies this table and is the primary key for this table.

Primary Key

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347



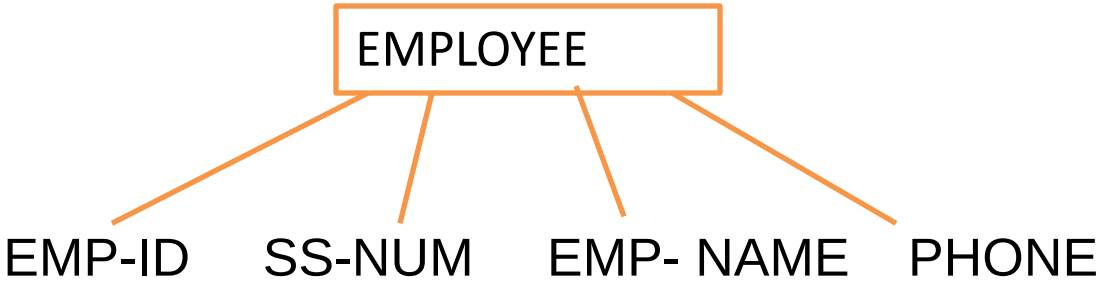
- Primary Key is the column you choose to maintain uniqueness in a table. In this table you can choose either Roll_No or PAN_No columns.
- Roll_No is preferable choice, as you cannot share PAN_No with everyone and there isn't any fixed sequence for PAN_No.

Composite Primary Key

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347

Roll_No and PAN_No makes up the primary key for this table. This is what is known as a **Composite Primary key**, that is, primary key that is made up of more than one field

Candidate Keys/Primary Key:



- Assume every employee has a phone#, only one phone# , and must have a phone# and that no two employees share the same phone#.
- What is(are) the Candiadate Key(s)?
- What would you choose as the Primary Key of table EMPLOYEE?

Primary Key:

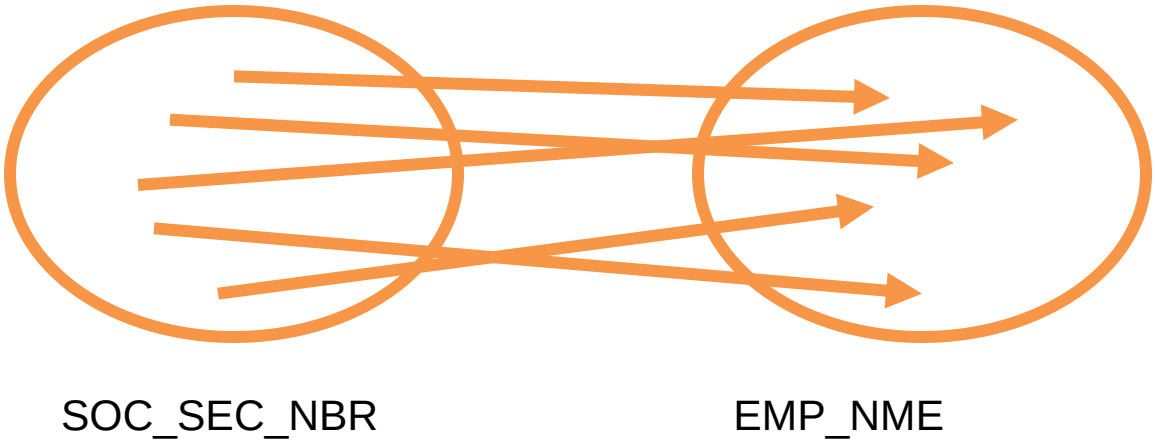
- The Primary Key MUST of course be a Determinant

  - i.e., all the other non-key attributes of a table must be functionally dependent on the primary key.

- In other words, for any given value of the primary key, one should get one and only value of the one non-key attributes

Functional Dependency

- Example
  - SOC_SEC_NBR → EMP_NME



-One and only one EMP_NME for a specific SOC_SEC_NBR

- SOC_SEC_NBR is the determinant of EMP_NME
- EMP_NME is functionally dependent on SOC_SEC_NBR

Determinants and Keys

What is (are) the determinant (s)?

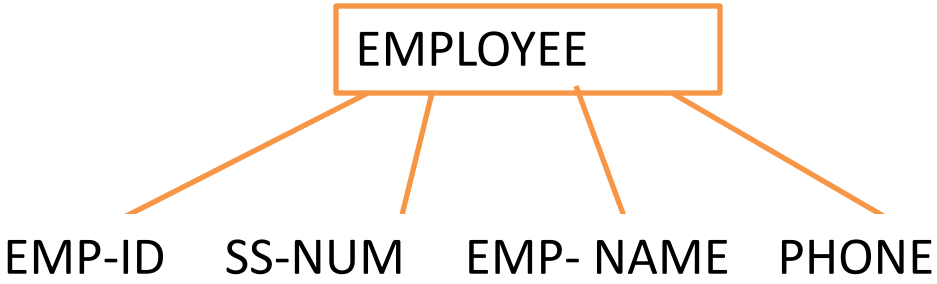
What is (are) the candidate key (s)?

What is the primary key?

Table: Student-Dorm-Fee

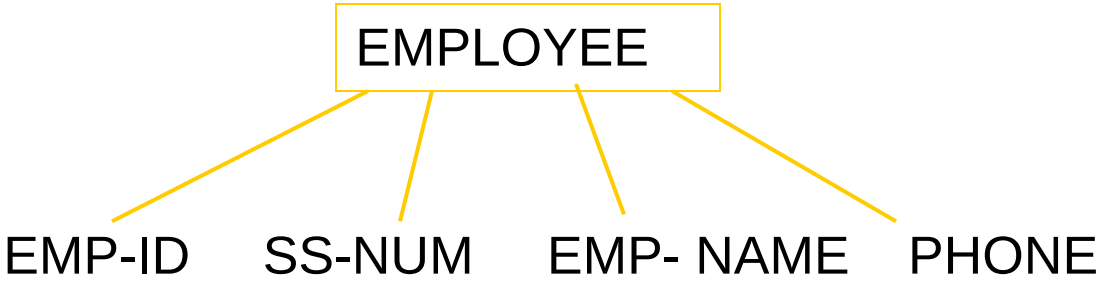
SID	DORM	FEE
101	Oracle	1000
102	Oracle	1000
103	DB2	800
104	DB2	800
105	Sybase	500

Primary Key



- Assume every employee must have a phone# , can have more than one phone #, and more than one employee share the same phone#.
- What is the Primary Key?

Primary Key



- Assume every employee must have a phone# , can have more than one phone #, but no two employees can share the same phone#.
- What is the Primary Key?

Primary Key:

```
SQL> CREATE TABLE dept
2.       (DEPT_NO int NOT NULL,
3.       DEPT_NAME varchar(20),
4.       LOCATION varchar(20),
5.       PRIMARY KEY (DEPT_NO));
```

```
SQL> ALTER TABLE dept ADD PRIMARY KEY (DEPT_NO);
```

Removing Primary Key:

```
SQL> ALTER TABLE dept DROP PRIMARY KEY;
```

3. Foreign Keys:

- A Foreign key is a key borrowed from another related table (that's why its foreign) in order to make the relationship between two tables.
- It is normally denoted with its first two letters, namely, FK

Foreign Keys:

- Foreign key – a (simple or composite) column which refers to the primary key of some table in a database.
- Foreign and primary keys must be defined on same data type.
- A foreign key may be contained in a primary key or another foreign key.

Foreign Keys:

student

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111

internships

Companies	Name	Roll_No
Microsoft	AAKUN GARG	Y12UC001
Google	AAKUN GARG	Y12UC001
Facebook	AAYUSH KUMAR	Y12UC002
NCR Corp	ABHILAKSHYA BHATEJA	Y12UC005

Foreign Key:

```
SQL> CREATE TABLE internships
2.      ( Companies varchar(20),
3.      Name varchar(20),
4.      Roll_No int NOT NULL,
5.      FOREIGN KEY (Roll_No)
6.      REFERENCES students(Roll_No));
```

Foreign Key:

```
SQL>  ALTER TABLE internships
2.        ADD FOREIGN KEY (Roll_No)
3.        REFERENCES students(Roll_No);
```

Removing Foreign Key:

```
SQL> ALTER TABLE internships DROP FOREIGN KEY;
```

Entity Integrity
and
Referential Integrity

Entity Integrity

- Entity Integrity is the mechanism the system provides to maintain primary keys.
- Entity Integrity – If the primary key (PK) is a composite key then all columns of the primary key must be non-null.
- The primary key is the only guaranteed way to positively identify rows in queries.

Entity Integrity:

- Entity Integrity ensures two properties for primary keys:
  - The primary key for a row is unique; it does not match the primary key of any other row in the table.
  - The primary key is not null, no component of the primary key may be set to null.
- The system enforces Entity Integrity by not allowing operations (INSERT, UPDATE) to produce an invalid primary key. Any operation that creates a duplicate primary key or one containing *nulls* is rejected.

Referential Integrity:

- Referential Integrity is the mechanism the system provides to maintain foreign keys
- Referential Integrity means that the Foreign key must match in terms of actual values and data types with the related Primary Key.

Rationale for Referential Integrity:

- Any non-primary key column may be unknown or inapplicable (wholly null).

- An unmatched non-null foreign key identifies a non-existent object and is in error

Referential Integrity Rules (Foreign Key Rules)

How is referential integrity maintained in a database?

Some operations that may cause a violation …

- Insert of PK values – no problem
- Update of PK values – what happens to matching foreign keys?
- Delete of PK values – what happens to matching foreign keys?
- Insert of FK values – disallowed unless matching primary key exists
- Update of FK values – disallowed unless matching primary key exists
- Delete of FK values (FK Values set to NULL) – no problem as long as NULL values are allowed in the FK

4. Alternate Key:

- Candidate column other the Primary column is the Alternate Key .
- May have null values.

Alternate Key:

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347

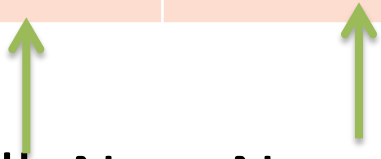
- If Roll_No is PK then PAN_No can be an Alternate key

5. Superkey:

- A superkey is any combo of columns on a table that can uniquely identify each and every row in a table.
- Add any other column/attribute to a Primary Key then it becomes a super key.

SuperKey:

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111
Y12UC022	AKASH GUPTA	ABC007	BH-5	347



- Roll_No + Name can be SuperKey

The SET UNUSED Option:

- Use SET UNUSED option to mark one or more columns as unused.
- Use DROP UNUSED COLUMNS option to remove the columns that are marked unused.

```
SQL> ALTER TABLE internships SET UNUSED (ROLL_NO);
```

```
SQL> ALTER TABLE internships DROP UNUSED COLUMNS;
```

Constrains:

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.
- Some valid constraints are:
  - NOT NULL
  - UJNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

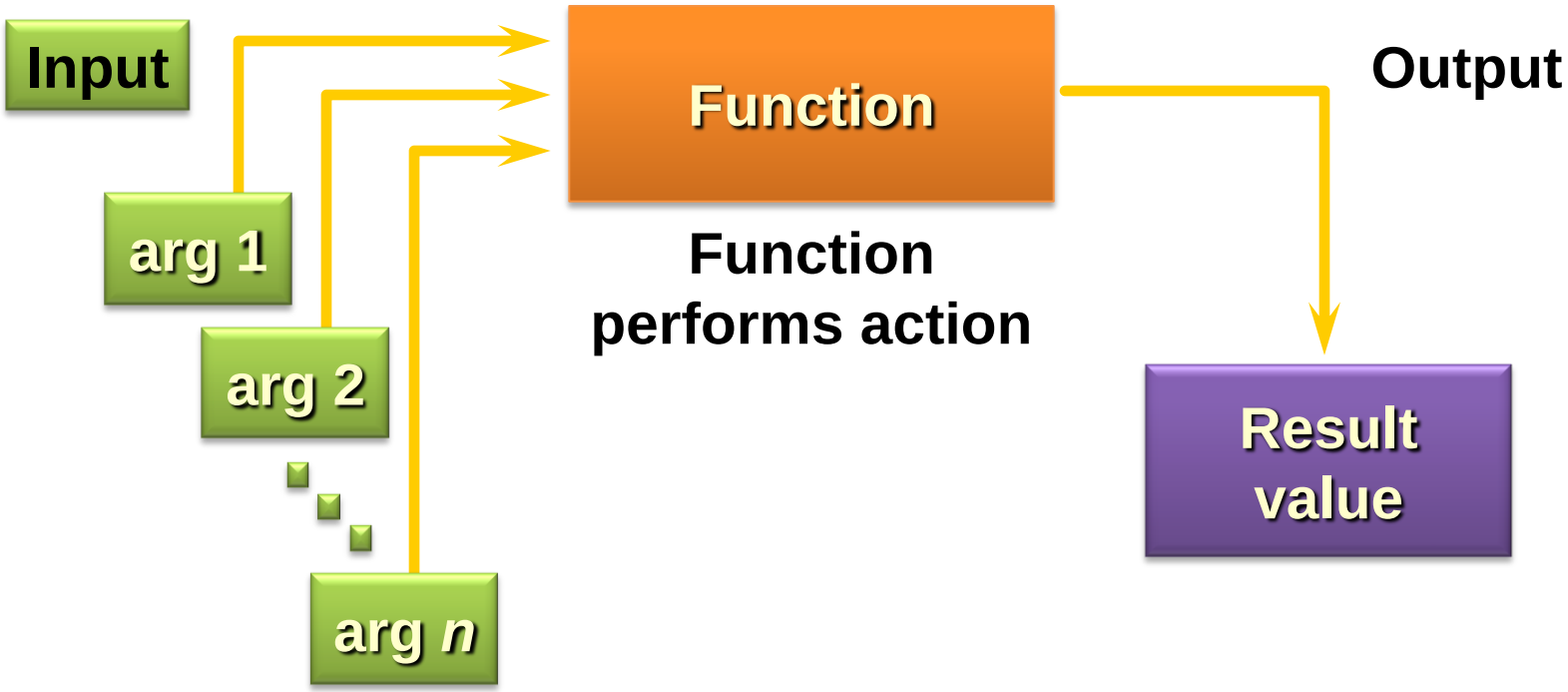
Constraints guidelines:

- Create a constraint:
  - At the same time as the table is created
  - After the table has been created
- Define a constraint at the column or table level.
- View the constraint in the data dictionary.

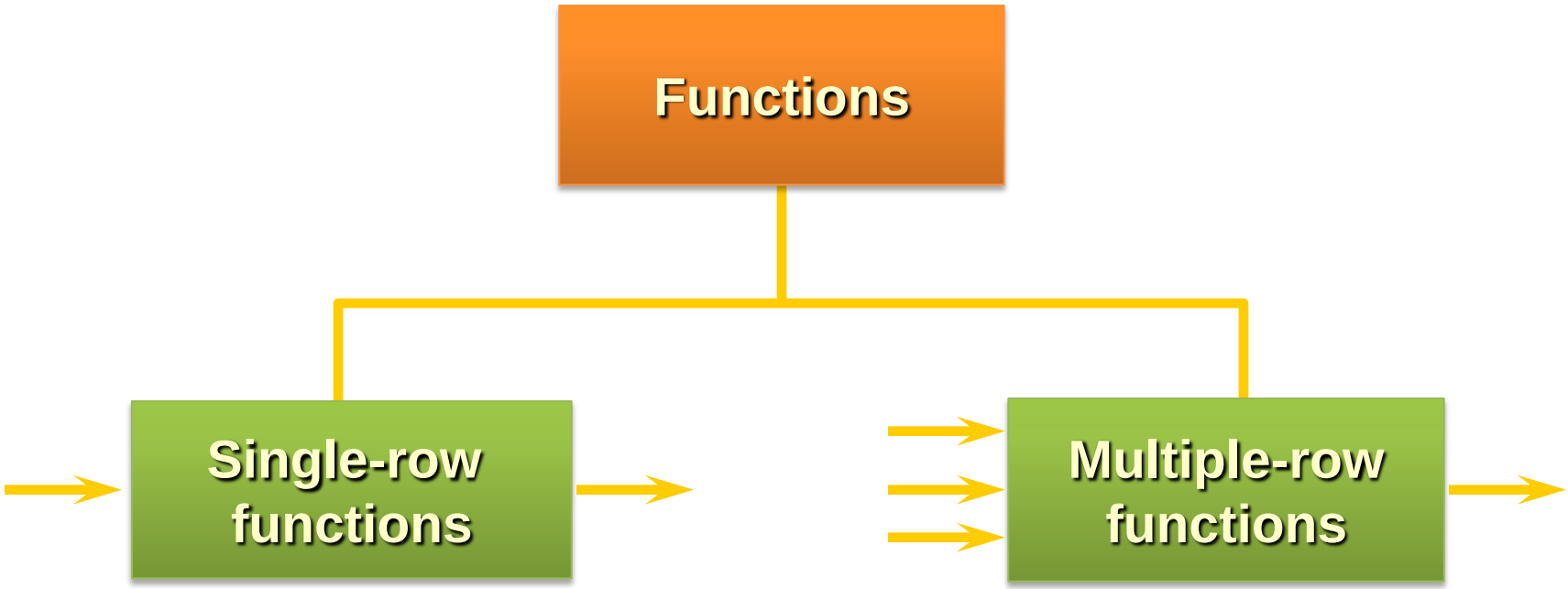
Possibilities with constraints:

- Adding a constraint
- Dropping a constraint
- Disabling a constraint
- Enabling a constraint
- Viewing constraint

SQL Functions



Two Types of SQL Functions:

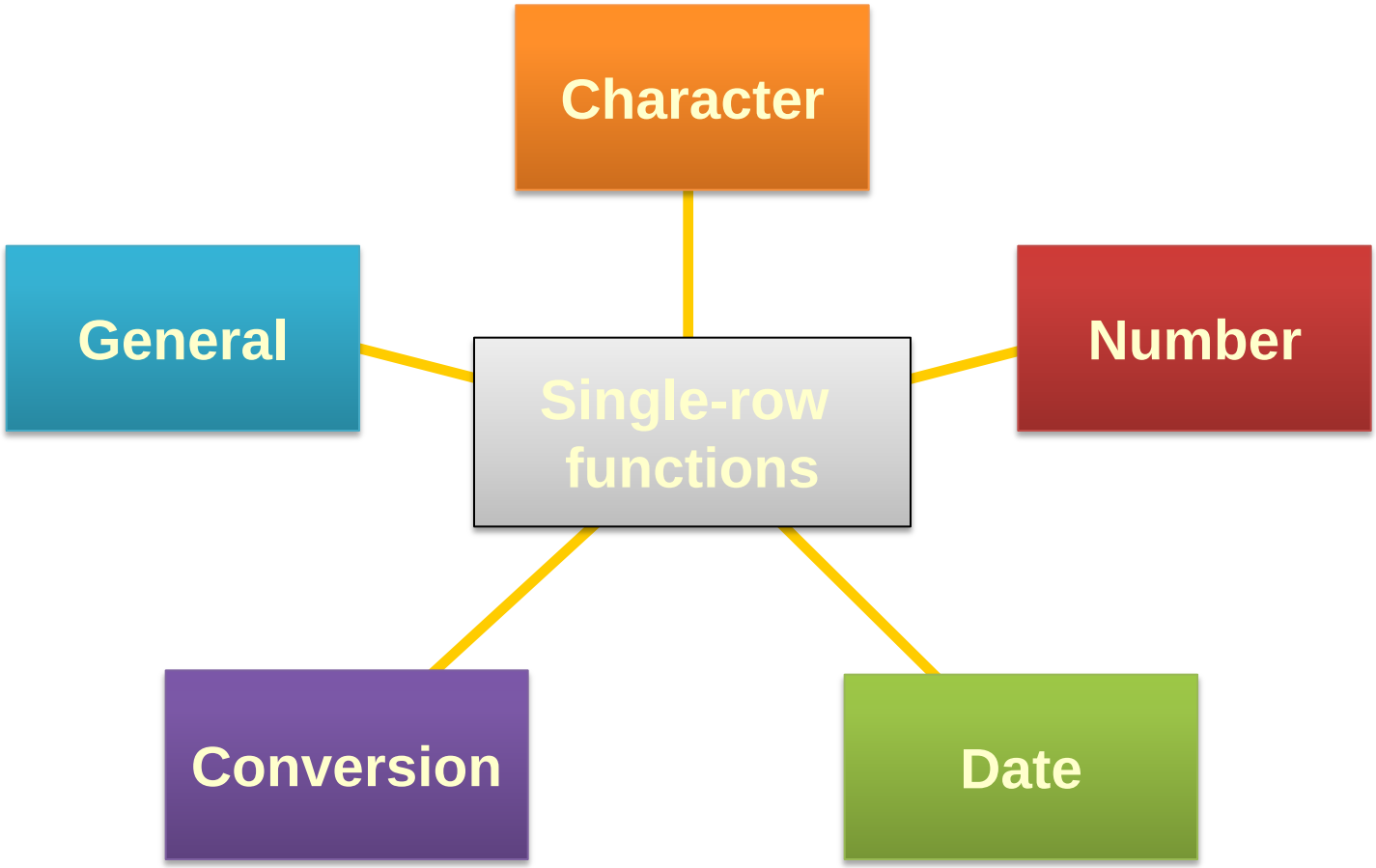


Single-Row Functions:

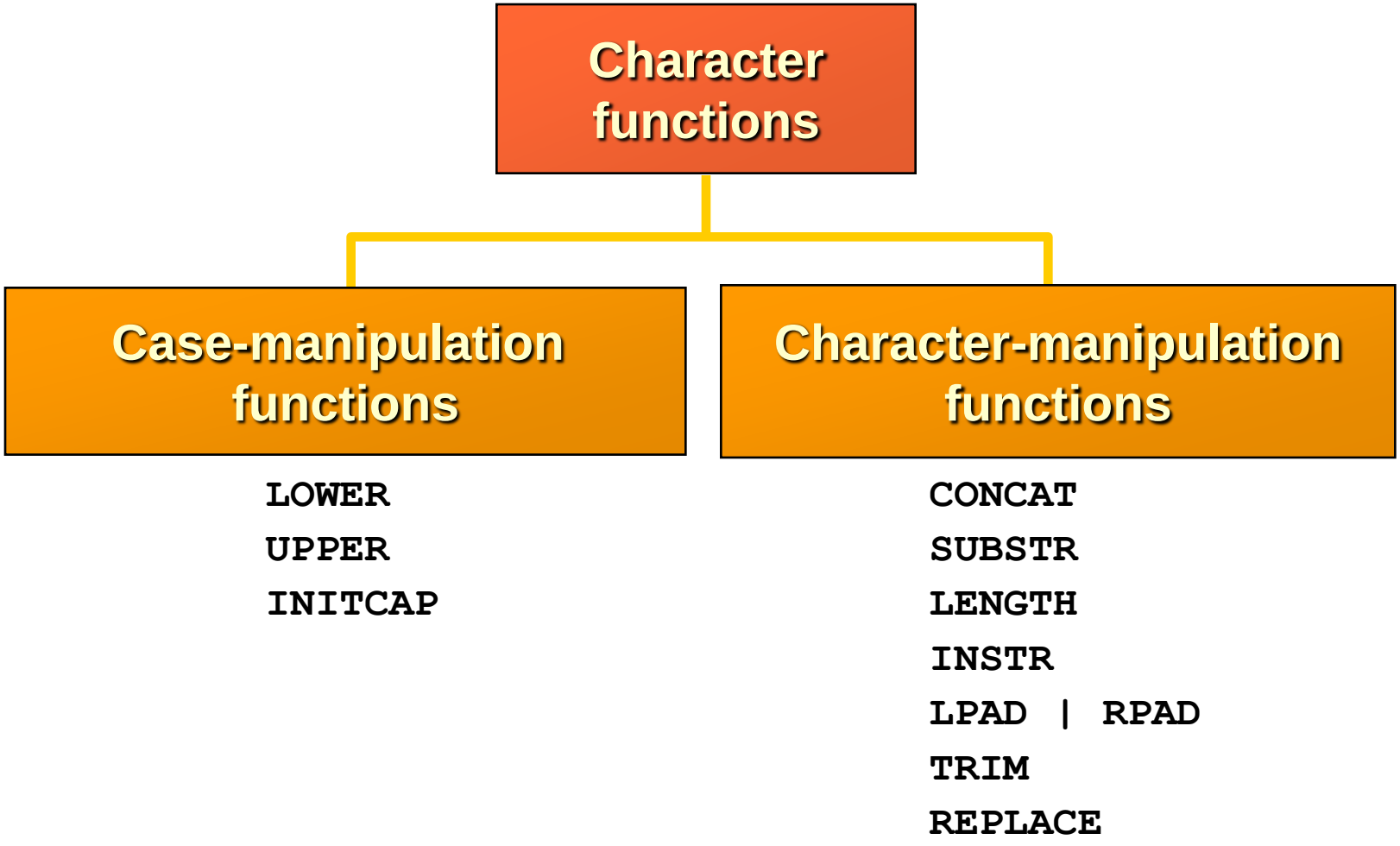
- Manipulate data items
- Accept arguments and return one value
- Act on each row returned
- Return one result per row
- May modify the data type
- Can be nested
- Accept arguments which can be a column or an expression

```
SQL> function_name [(arg1, arg2,...)]
```

Single-Row Functions



Character Functions



Number Functions

- `ROUND`: Rounds value to specified decimal

  `ROUND(45.926, 2)` → `45.93`

- `TRUNC`: Truncates value to specified decimal

  `TRUNC(45.926, 2)` → `45.92`

- `MOD`: Returns remainder of division

  `MOD(1600, 300)` → `100`

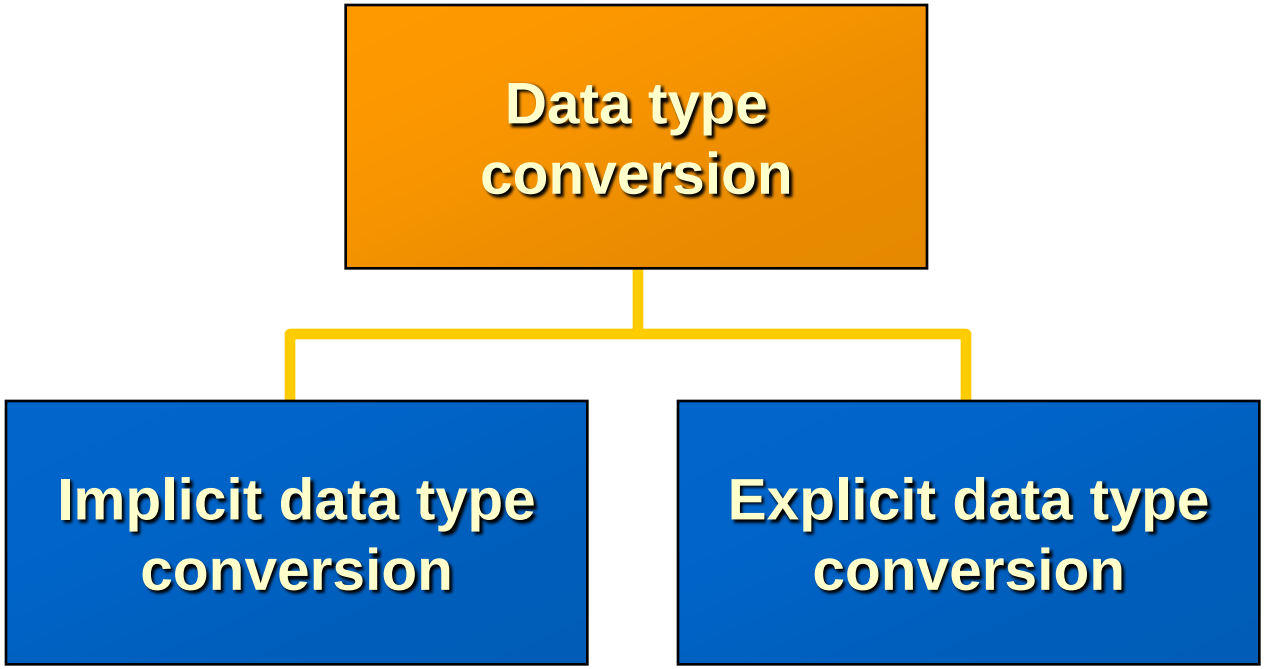
Date Functions

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Next day of the date specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date

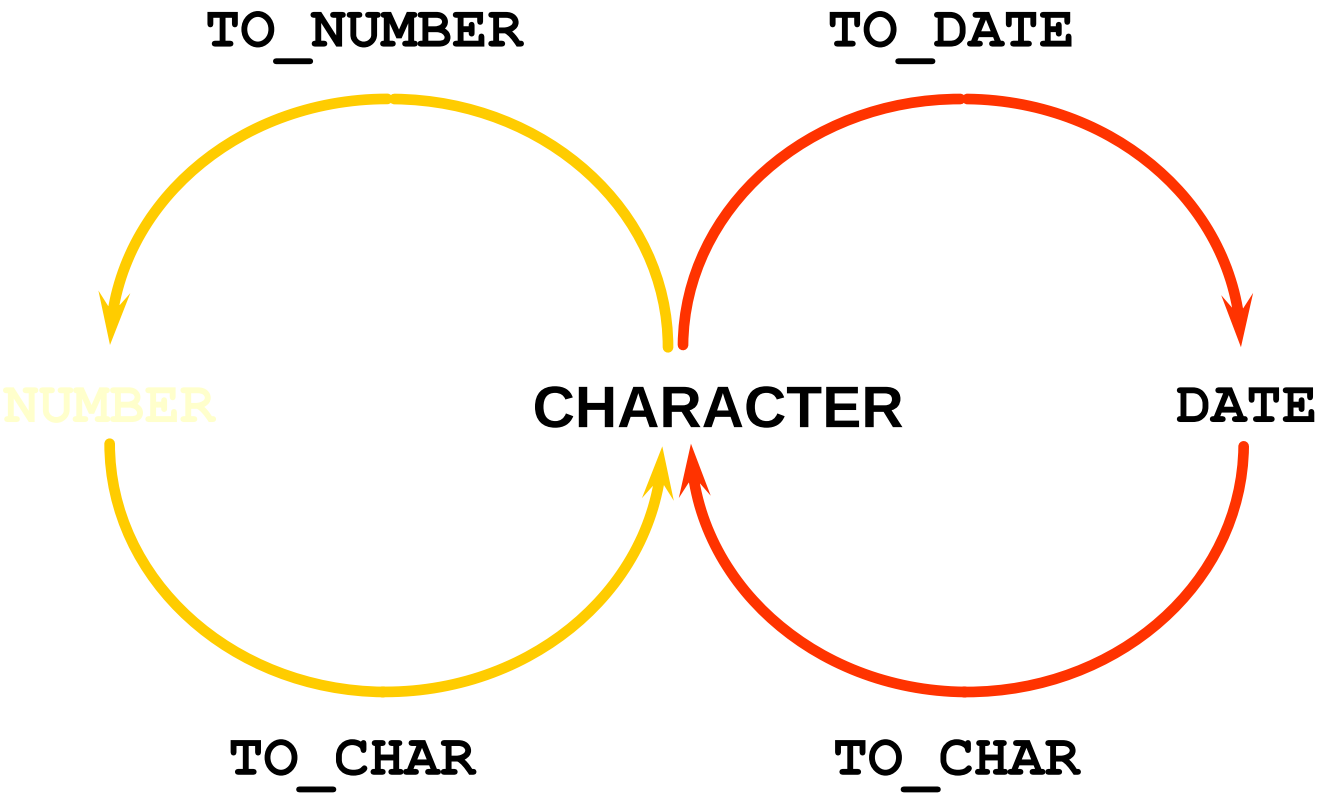
Using Date Functions

- `MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')` → `19.6774194`
- `ADD_MONTHS ('11-JAN-94',6)` → `'11-JUL-94'`
- `NEXT_DAY ('01-SEP-95','FRIDAY')` → `'08-SEP-95'`
- `LAST_DAY('01-FEB-95')` → `'28-FEB-95'`

Conversion Functions

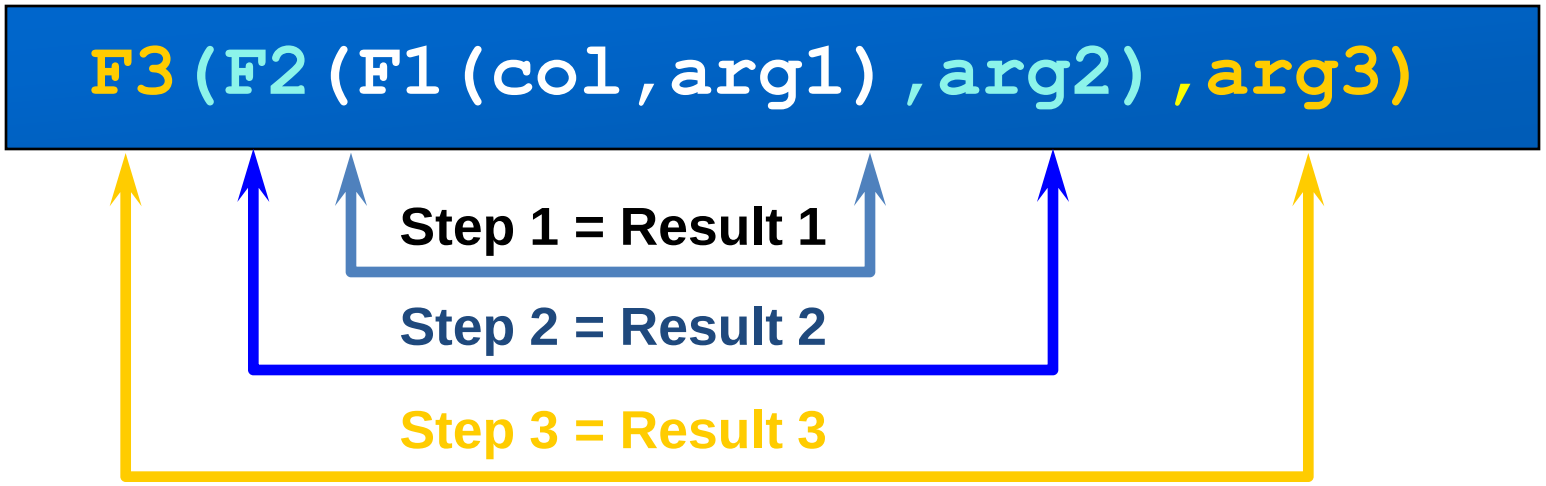


Explicit Data Type Conversion



Nesting Functions

- Single-row functions can be nested to any level.
- Nested functions are evaluated from deepest level to the least deep level.



Group Functions:

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE

Displaying Data from Multiple Tables

Displaying Data from Multiple Tables

student

Roll_No	Name	PAN_No	Hostel	Room_No
Y12UC001	AAKUN GARG	ABC001	BH-1	101
Y12UC002	AAYUSH KUMAR	ABC002	BH-2	101
Y12UC005	ABHILAKSHYA BHATEJA	ABC003	BH-3	501
Y12UC012	ADITI GUPTA	ABC004	GH-1	510
Y12UC019	AGAM AGARWAL	ABC005	BH-4	501
Y12UC020	AISHWARYA GUPTA	ABC006	GH-2	111

internships

Companies	Name	Roll_No
Microsoft	AAKUN GARG	Y12UC001
Google	AAKUN GARG	Y12UC001
Facebook	AAYUSH KUMAR	Y12UC002
NCR Corp	ABHILAKSHYA BHATEJA	Y12UC005

Join:

- Use a join to query data from more than one table
- Write the join condition in the WHERE clause.
- Prefix the column name with the table name when the same column name appears in more than one table.

```
SQL>    SELECT    table1.column, table2.column
2.      FROM      table1, table2
3.      WHERE     table1.column1 = table2.column2;
```

Cartesian Product:

- A Cartesian product is formed when:
  - A join condition is omitted
  - A join condition is invalid
  - All rows in the first table are joined to all rows in the second table
- To avoid a Cartesian product, always include a valid join condition in a WHERE clause.