

Keys

- Consist of one or more attributes that determine other attributes
- Used to:
 - Ensure that each row in a table is uniquely identifiable
 - Establish relationships among tables and to ensure the integrity of the data
- Primary key (PK):** Attribute or combination of attributes that uniquely identifies any given row

Types of Keys

- Null:** Absence of any data value that could represent:
 - An unknown attribute value
 - A known, but missing, attribute value
 - A inapplicable condition
- Referential integrity:** Every reference to an entity instance by another entity instance is valid

Dependencies

- Functional dependence:** Value of one or more attributes determines the value of one or more other attributes
 - Determinant:** Attribute whose value determines another
 - Dependent:** Attribute whose value is determined by the other attribute
- Full functional dependence:** Entire collection of attributes in the determinant is necessary for the relationship

Ways to Handle Nulls

- Flags:** Special codes used to indicate the absence of some value
- NOT NULL constraint:** Placed on a column to ensure that every row in the table has a value for that column
- UNIQUE constraint:** Restriction placed on a column to ensure that no duplicate values exist for that column

Types of Keys

- Composite key:** Key that is composed of more than one attribute
- Key attribute:** Attribute that is a part of a key
- Entity integrity:** Condition in which each row in the table has its own unique identity
 - All of the values in the primary key must be unique
 - No key attribute in the primary key can contain a null

Relational Set Operators

Select (Restrict)

- Unary** operator that yields a horizontal subset of a table

Project

- Unary** operator that yields a vertical subset of a table

Union

- Combines all rows from two tables, excluding duplicate rows
- Union-compatible:** Tables share the same number of columns, and their corresponding columns share compatible domains

Intersect

- Yields only the rows that appear in both tables
- Tables must be union-compatible to yield valid results

Relational Set Operators

Difference

- Yields all rows in one table that are not found in the other table
- Tables must be union-compatible to yield valid results

Product

- Yields all possible pairs of rows from two tables

Relational Set Operators

Join

- Allows information to be intelligently combined from two or more tables

Divide

- Uses one 2-column table as the dividend and one single-column table as the divisor
- Output is a single column that contains all values from the second column of the dividend that are associated with every row in the divisor

Types of Joins

- Natural join:** Links tables by selecting only the rows with common values in their common attributes
- Join columns:** Common columns
- Equijoin:** Links tables on the basis of an equality condition that compares specified columns of each table
- Theta join:** Extension of natural join, denoted by adding a theta subscript after the JOIN symbol

Types of Joins

- Inner join:** Only returns matched records from the tables that are being joined
- Outer join:** Matched pairs are retained and unmatched values in the other table are left null
- Left outer join:** Yields all of the rows in the first table, including those that do not have a matching value in the second table
- Right outer join:** Yields all of the rows in the second table, including those that do not have matching values in the first table

Right outer join, full outer join

Output all rows of the right (AGENT) table, including ones for which there are no matching values in the join column in the other (CUSTOMER) table:

AGENT	CUSTOMER
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50

Outer joins are useful in exposing missing information [in our example, customers who don't seem to have an agent, and, agents who don't seem to have customers].

A 'full outer join' is a union of left outer join and right outer join - output all the rows from both tables, including ones for which there are no matches in the other table - this could result in nulls on the left side of some rows, as well as nulls on the right side of others.

Data Dictionary and the System Catalog

- Data dictionary** ^{Metadata}: Description of all tables in the database created by the user and designer
- System catalog**: System data dictionary that describes all objects within the database
- Homonyms and synonyms must be avoided to lessen confusion
 - Homonym**: Same name is used to label different attributes
 - Synonym**: Different names are used to describe the same attribute