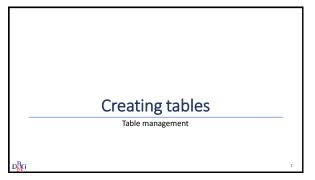


The SQL Language: Table management ➤ Creating a table ➤ Altering a table ➤ Deleting a table ➤The data dictionary ➤ Data integrity  $D_{MG}^{BG}$ 

1

3



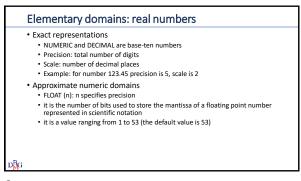
CREATE TABLE TableName (AttributeName Domain [DefaultValue] [Constraints] {, AttributeName Domain [DefaultValue] [Constraints]} OtherConstraints It allows
 defining all attributes (i.e., columns) of the table
 defining integrity constraints on the table data Userning.
Domain
 it defines the data type of an attribute
 predefined domains of the SQL language (elementary domains)
 user-defined domains (starting from the predefined domains) CREATE Constraints
 integrity constraints for the specific attribute
 OtherConstraints
 general integrity constraints on the table

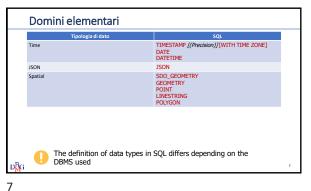
2

Domain definitions • Default • it allows specifying a default value for the attribute • GenericValue a value compatible with the attribute domain • \*USER • user identifier • NULL · standard default value DEFAULT < GenericValue | USER | CURRENT\_USER | SESSION\_USER | SYSTEM\_USER | NULL>

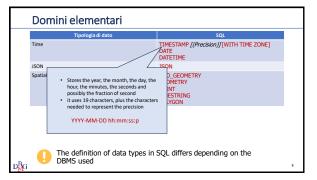
**Elementary domains** CHARACTER [VARYING] [(Length)]
[CHARACTER SET CharacterFamilyName]
VARCHAR (Length)
TEXT BIT [VARYING] [(*Length*)] BLOB BINARY BOOLEAN Integer numbers INTEGER SMALLINT BIGINT NUMERIC [( Precision, Scale )]
DECIMAL [( Precision, Scale )] Real numbers FLOAT [(n)]
REAL
DOUBLE PRECISION

5





6



Defining a domain (1/2)

CREATE DOMAIN DomainName AS DataType

[DefaultValue][Constraint]

• It defines a new domain that may be used in attribute definitions

• DataType is an elementary domain

• Example

CREATE DOMAIN Grade AS SMALLINT

DEFAULT NULL

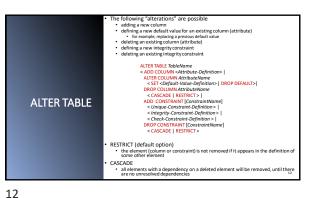
CHECK (Grade >= 18 and Grade <=30)

9

8

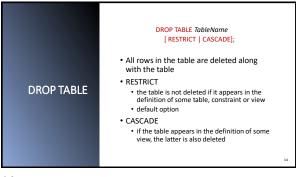
Definition of supplier-product database				
Creating the Supplier Table CREATE TABLE S (     SId CHAR(5),     SName CHAR(20),     NEmployees SMALLINT,     City CHAR(15));      Creating the Supply Table CREATE TABLE SP (		Creating the CREATE TABL PId PName Color Size Store	e Product Table E P ( CHAR(6), CHAR(20), CHAR(6), SMALLINT, CHAR(15));	
SId PId Qty D <mark>R</mark> G	CHAR(5), CHAR(6), INTEGER);	The definition	on of integrity constraints is missing	





**Deleting tables** Table management D<mark>B</mark>G

13



**Data Dictionary** 

15 14

### The data dictionary · Metadata are information (data) about data they may be stored in database tables • The data dictionary contains the metadata of a relational database · it contains information about the database objects • it is managed directly by the relational DBMS . it may be gueried by means of SQL commands • It contains various information descriptions of all database structures (tables, indices, views) SQL stored procedures user privileges statistics on the database tables on the database indices on the database views on the evolution of the database

Information about tables • For each database table, the data dictionary contains • table name and physical structure of the file storing the table • name and data type for each attribute • name of all indices created on the table • integrity constraints

### Data dictionary tables

- Data dictionary information is stored in several tables • each DBMS uses different names for different tables
- The data dictionary may be queried by means of SQL commands

₽<mark>B</mark>G

18

### The Oracle data dictionary

- In Oracle 3 collections of information are defined for the data dictionary
  - USER\_\*: metadata related to the current user's data
    ALL\_\*: metadata related to all users' data
  - ALL\_\*: metadata related to all users: data
  - DBA\_\*: metadata about system tables
- USER\_\* contains different tables and views, including:
  - USER\_TABLES contains metadata to the user tables
  - USER\_TAB\_STATISTICS contains statistics computed on the user tables
- USER\_TAB\_COL\_STATISTICS contains statistics computed on user table columns

р<mark>В</mark>С

Querying the data dictionary no. 1

• Show the name of user-defined tables and the number of tuples stored in each table

SELECT Table\_Name, Num\_Rows FROM USER\_TABLES;

 Table\_Name
 Num\_Rows

 S
 5

 P
 6

 SP
 12

20

Querying the data dictionary no.2 (1/2)

 For each attribute in the supplier-product table, show the attribute name, the number of distinct values and the number of tuples with a NULL value

> SELECT Column\_Name, Num\_Distinct, Num\_Nulls FROM USER\_TAB\_COL\_STATISTICS WHERE Table\_Name = 'SP' ORDER BY Column Name;

Colu

Column_Name	Num_Distinct	Num_Nulls
SId	4	0
PId	6	0
Qty	4	0

21

### **Data Integrity**

Table management

22

### • Data in a database are corr

- Data in a database are correct if they satisfy a set of correctness rules
  - rules are called *integrity constraints*
  - example: Qty >=0
- Data update operations define a new state for the database, which may not necessarily be correct
- Checking the correctness of a database state may be done
  - by application procedures, performing all required checks
  - through the definition of *integrity constraints* on the tables
  - through the definition of triggers

рВс

## Application procedures Each application includes all required correctness checks Pros Cons • "flexible" approach • checks may be "circumvented" by interacting directly with the DBMS • a coding error may have significant effects on the database • the knowledge about integrity constraints is typically "hidden" inside applications

D<mark>B</mark>G

### Table integrity constraints

- · Integrity constraints are
  - defined in the CREATE or ALTER TABLE statements
  - · stored in the system data dictionary
- Each time data are updated, the DBMS automatically verifies that the constraints are satisfied

₽<mark>B</mark>G

24 25

### Pros declarative definition of constraints, whose verification is delegated to the system the data dictionary describes all of the constraints in in the system unique centralized check point constraint verification may not be circumvented the constraint verification may not be circumvented the constraint verification may not be circumvented they may slow down application execution tis is not possible to define constraints of an arbitrary type example: constraints on aggregated data

Table integrity constraints

### Triggers

- Triggers are procedures executed automatically when specific data updates are performed
  - defined through the CREATE TRIGGER command
  - stored in the system data dictionary
- When a modification event occurs on data under the trigger's control, the procedure is automatically executed

LN

26 27

# Pros • they allow defining complex constraints • normally used in combination with constraint definition on the tables • unique centralized check point • constraint verification may not be circumvented • complex • they may slow down application execution

### Fixing violations

- If an application tries to execute an operation that causes a constraint violation, the system may
  - block the operation, causing an error in the application execution
  - execute a compensating action so that a new correct state is reached
    - example: when a supplier is deleted, its supplies are also deleted

υį

### Integrity constraints in SQL-92

- The SQL-92 standard introduced the possibility to specify integrity constraints in a declarative way, delegating to the system the verification of their consistency
  - · table constraints
  - restrictions on the data allowed in table columns
  - referential integrity constraints
    - manage references among different tables
       based on the concept of foreign key

 $D_{MG}^{BG}$ 30

### **Table Constraints**

- They are defined on one or more columns of a table
- They are defined in the creation instructions of:
  - TablesDomains
- · Type of constraints:
  - Primary key
     Admissibility of NULL values
  - Uniqueness
  - General tuple constraints
- They are checked after each SQL statement that operates on the table subject to the constraint
  - Entering new data Changing the value of constrained columns
- If a constraint is violated, the SQL statement that caused the violation results in an execution error

31

33

• A primary key is a set of attributes that uniquely identifies rows in a tables • Only one primary key may be specified for a given table · Primary key definition composed of a single attribute **Primary Key** AttributeName Domain PRIMARY KEY • composed of one or more attributes PRIMARY KEY (ListOfAttributes)

**Primary Key examples** a single attribute CREATE TABLE S ( CHAR(5) PRIMARY KEY, SId SName CHAR(20), SMALLINT, NEmployees CHAR(15)) one or more attributes CREATE TABLE SP ( CHAR(5), CHAR(6), SId PId Qty INTEGER, PRIMARY KEY (SId, PId));

32

Admissibility of the NULL value

- The NULL value indicates absence of
- When a value must always be specified for a given attribute

AttributeName Domain NOT NULL

· NULL value is not allowed

NOT NULL: example CREATE TABLE S (SId CHAR(5), SName CHAR(20) NOT NULL, NoEmployees SMALLINT, City CHAR(15));

34 35

· An attribute or a set of attributes may not take the same value in different rows of the table • for a single attribute AttributeName Domain UNIQUE UNIQUE · for one or more attributes UNIQUE (ListOfAttributes) • It is possible to repeat the NULL value (it is seen as a different value in each row) 36

Candidate key • A candidate key is a set of attributes that may serve as a primary key • it is unique · it does not allow the NULL value • The combination UNIQUE NOT NULL defines a candidate key that does not allow null values AttributeName Domain UNIQUE NOT NULL

D<mark>B</mark>G

37

### Unique constraint: example CREATE TABLE P ( PId CHAR(6), CHAR(20) NOT NULL UNIQUE, **PName** Color CHAR(6), SMALLINT, Size CHAR(15)); Store

• They allow expressing general conditions on each tuple • tuple or domain constraints General Tuple AttributeName Domain CHECK (Condition) Constraints • Predicates that can be specified in the WHERE clause can be specified as a condition • The database is correct if the condition is true

38 39

### General tuple constraints: example CREATE TABLE S (SId CHAR(5) PRIMARY KEY, SName CHAR(20) NOT NULL, SMALLINT NoEmployees CHECK (NoEmployees>0), City CHAR(15));

• They manage the link between tables by means of the value of attributes The foreign key is defined in the CREATE TABLE Referential statement of the referencing table Integrity Constraints FOREIGN KEY (ListReferencingAttributes ) REFERENCES TableName [(ListReferencedAttributes)] • If the referenced attributes have the same name as the referenced attributes, they are not required

40 41

### Example: Defining a Foreign Key CREATE TABLE SP ( CHAR(5), CHAR(6), PId INTEGER. Otv PRIMARY KEY (SId, PId), FOREIGN KEY (SId) REFERENCES S(SId). FOREIGN KEY (PId) REFERENCES P(PId));

Integrity constraints are checked after each SQL command that may cause their violation Insert or update operations on the referencing table that violate the constraints are not allowed In the CREATE TABLE statement of the referencing table FOREIGN KEY (ListReferencingAttributes) TableName [(ListReferencedAttributes )] Politiche di [ON OPDATE

CASCADE | SET DEFAULT | SET NULL | NO ACTION>]

[ON DELETE

CASCADE | SET DEFAULT | SET NULL | NO ACTION>] gestione dei vincoli Update or delete operations on the referenced table have the following outcome on the referencing table: CASCADE: the update or delete operation is propagated

SET NULL/DEFAULT: a null or default value is set in the columns
for the tuples whose values are no longer present in the
referenced table NO ACTION: the offending action is not executed

42 43

### **Example: Product-Supply DB**

• table P: describes the available products

b₿G

b₿G

- Primary key: Pld
- Product name cannot have NULL or duplicate values
- The size is always greater than
- table S: describes suppliers
- Primary key: Sid
- · Supplier name cannot have NULL or duplicate values
- The number of employees is always greater than zero
- table SP: describes supplies, relating products to the suppliers who supply them
  - · Primary key: (SId, PId)
  - · Quantity cannot be null and is greater than zero
  - Referential integrity constraints

### Constraint Management: Example 1

- SP (referencing table)
- insert (new tuple Pld, Sld) -> No
- update (SId) -> No
- delete (tuple) -> Ok
- S (referenced table)
  - insert (new tuple) -> Ok • update (SId) -> cascaded update (cascade)
  - delete (tuple) -> cascaded update (cascade)

prevent action (no action)

45

44

### SQL Example: Product-Supply DB

CREATE TABLE SP (SId (Sid CHAR(6),
PId CHAR(6),
PID CHAR(6),
PID CHAR(6),
PRIMARY KE (SIG PId),
ON DELETE NO ACTION
ON UPDATE CASCADE,
FOREIGN KEY (PId)
ON DELETE NO ACTION
ON UPDATE CASCADE,
ON DELETE NO ACTION
ON UPDATE CASCADE, CHAR(6) PRIMARY KEY, CHAR(20) NOT NULL UNIQUE, CHAR(6), CREATE TABLE P ( PId PNam Color Size CHECK (Size > 0), CHAR(15)); CHAR(5) PRIMARY KEY, CHAR(20) NOT NULL UNIQUE, SMALLINT SName NoEmployees CHECK (NoEmployees>0), CHAR(15)); City

Constraint Management: Example 2

- Employees (Eld, EName, City, DId)
- Departments (DId, DName, City)
- · Employees (referencing table)
  - insert (new tuple) -> No update (DId) -> No delete (tuple) -> Ok
  - delete (tuple)
- Departments (referenced table)
  - insert (new tuple) -> Ok
  - update (DId) -> cascaded update (cascade) -> cascaded update (cascade) prevent action (no action) · delete (tuple)

set to unknown value (set null) set to default value(set default)