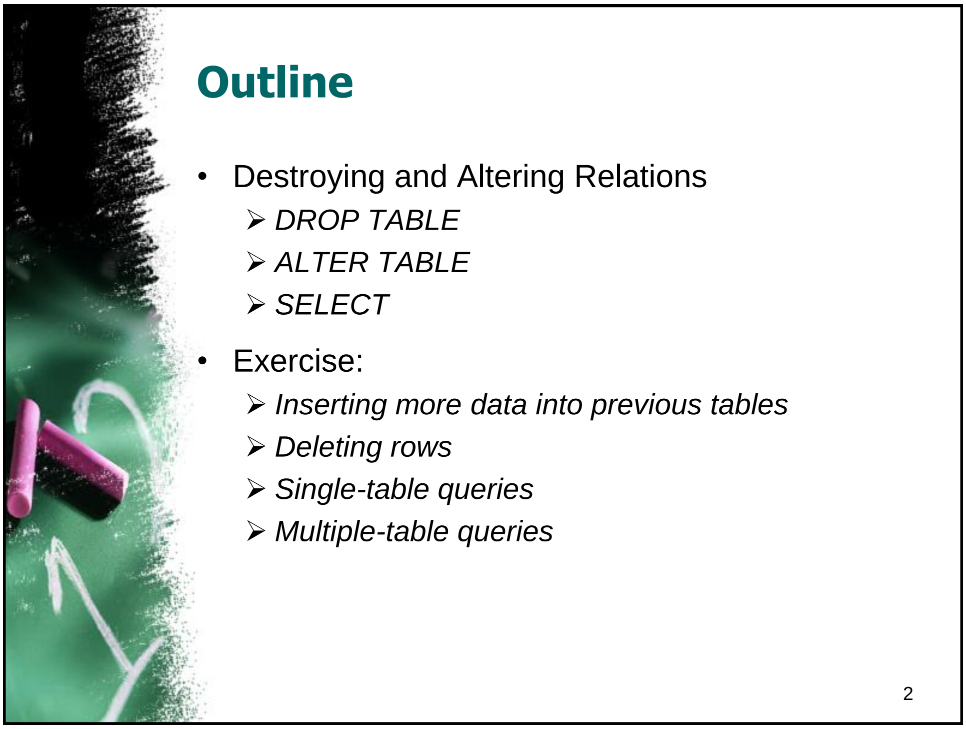
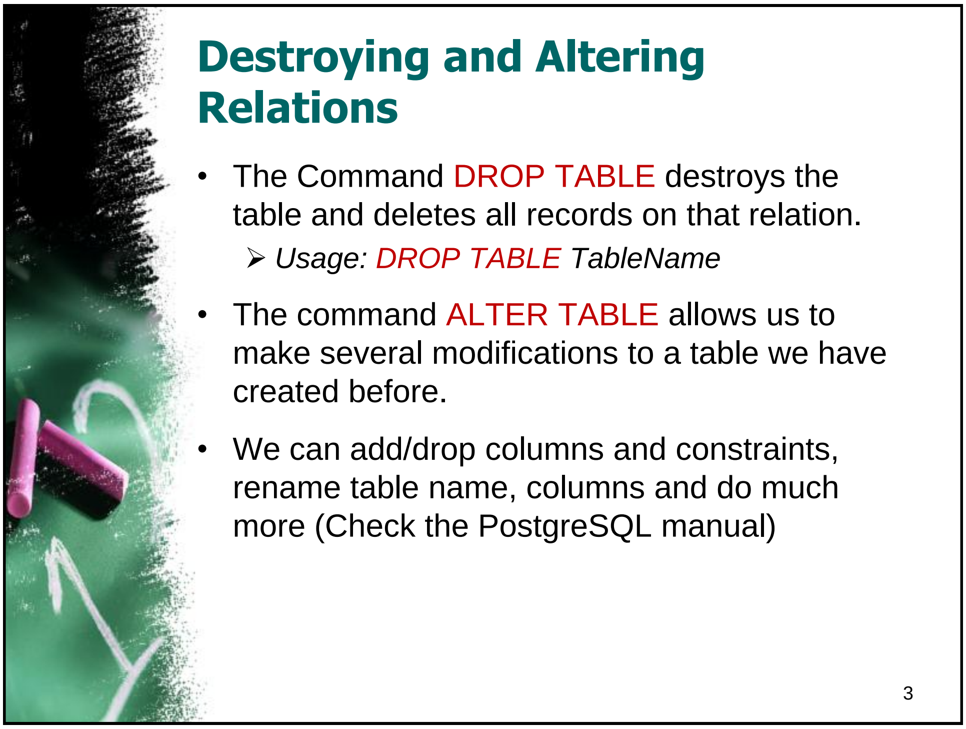
CSI 2132 Lab 3

More on SQL



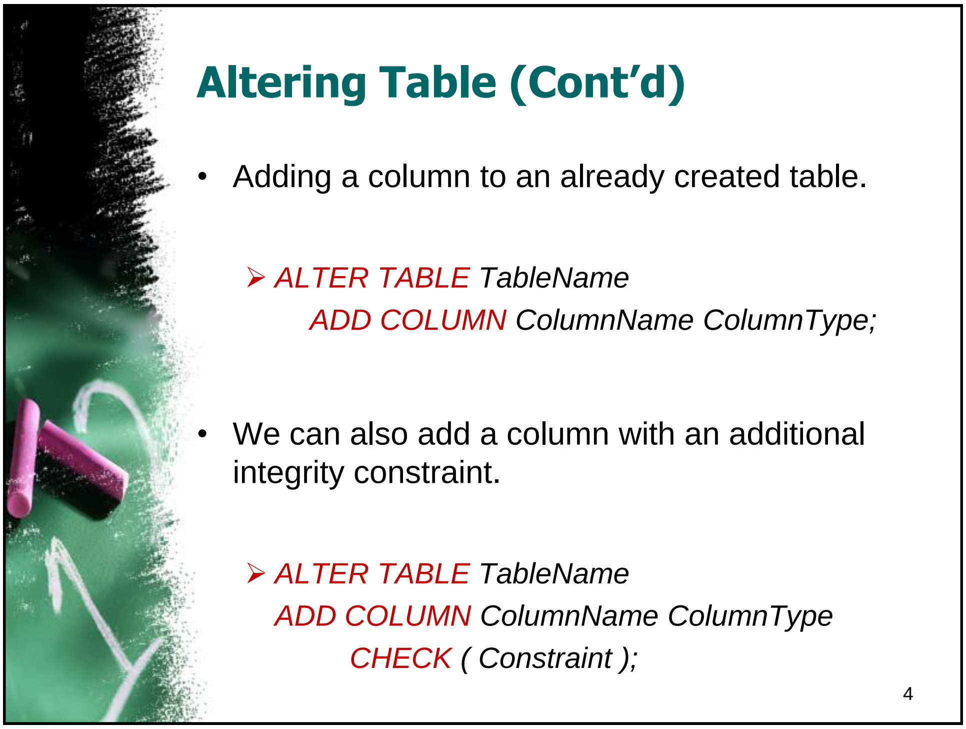
Outline

- Destroying and Altering Relations
  - *DROP TABLE*
  - *ALTER TABLE*
  - *SELECT*
- Exercise:
  - *Inserting more data into previous tables*
  - *Deleting rows*
  - *Single-table queries*
  - *Multiple-table queries*



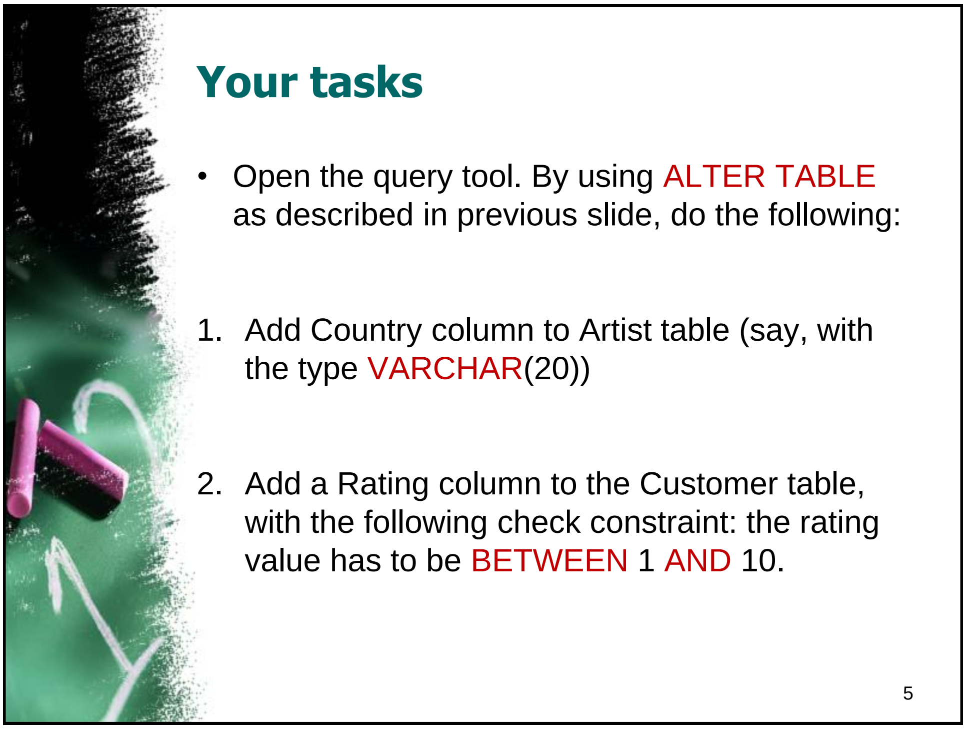
Destroying and Altering Relations

- The Command DROP TABLE destroys the table and deletes all records on that relation.
  - *Usage: DROP TABLE TableName*
- The command ALTER TABLE allows us to make several modifications to a table we have created before.
- We can add/drop columns and constraints, rename table name, columns and do much more (Check the PostgreSQL manual)



Altering Table (Cont’d)

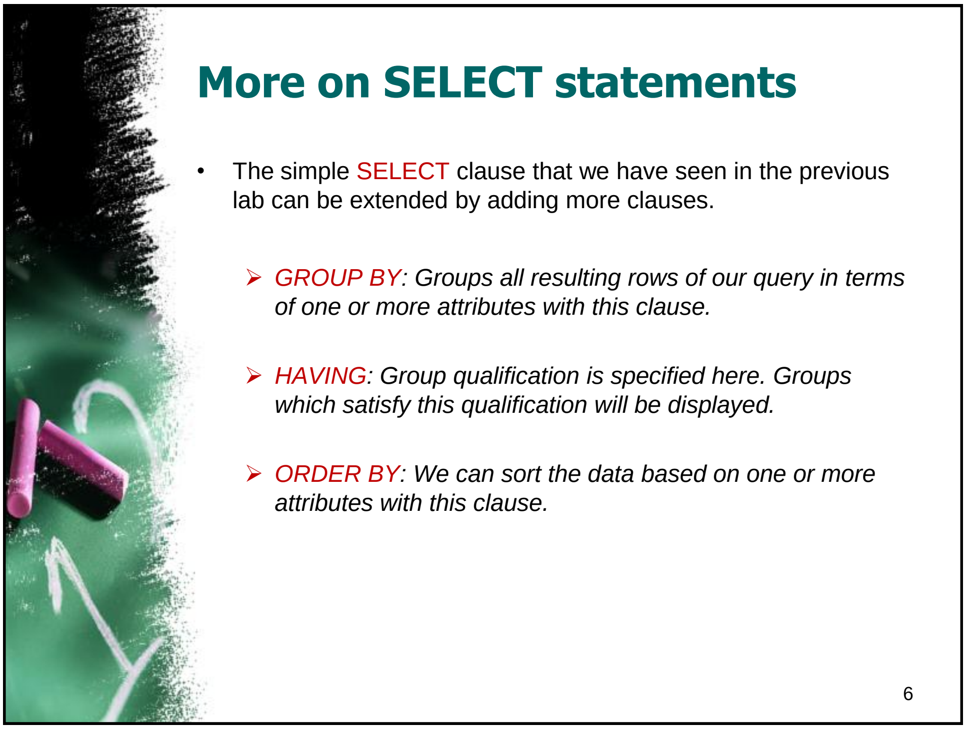
- Adding a column to an already created table.
  - *ALTER TABLE TableName*
    ADD COLUMN ColumnName ColumnType;
- We can also add a column with an additional integrity constraint.
  - *ALTER TABLE TableName*
    ADD COLUMN ColumnName ColumnType
    CHECK (Constraint);



Your tasks

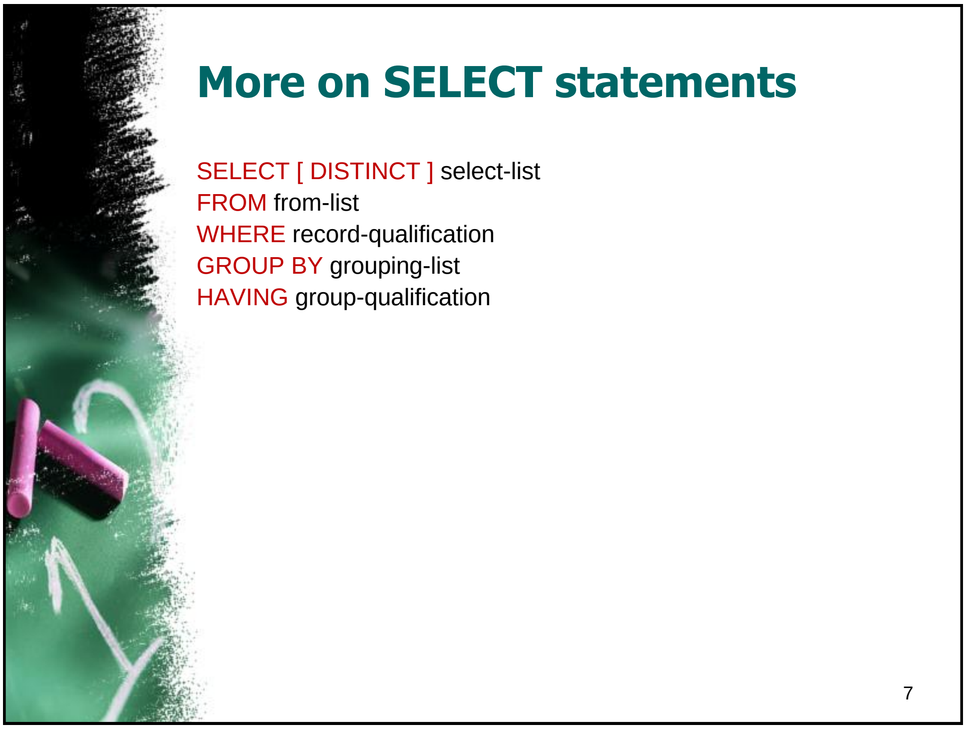
- Open the query tool. By using ALTER TABLE as described in previous slide, do the following:

1. Add Country column to Artist table (say, with the type VARCHAR(20))
2. Add a Rating column to the Customer table, with the following check constraint: the rating value has to be BETWEEN 1 AND 10.



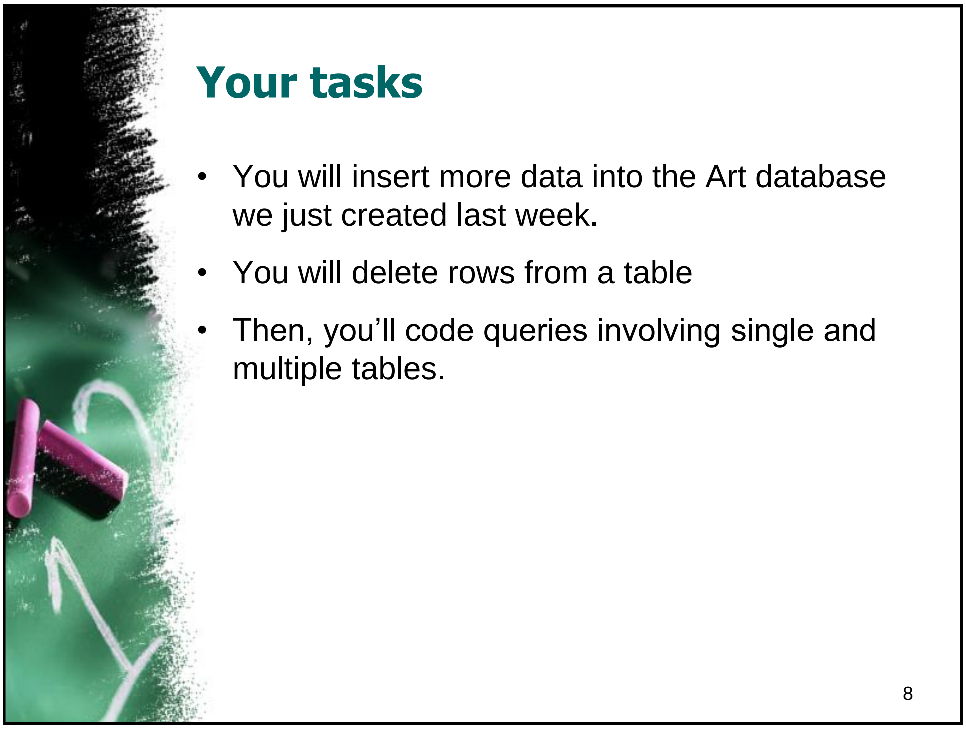
More on SELECT statements

- The simple SELECT clause that we have seen in the previous lab can be extended by adding more clauses.
  - *GROUP BY: Groups all resulting rows of our query in terms of one or more attributes with this clause.*
  - *HAVING: Group qualification is specified here. Groups which satisfy this qualification will be displayed.*
  - *ORDER BY: We can sort the data based on one or more attributes with this clause.*



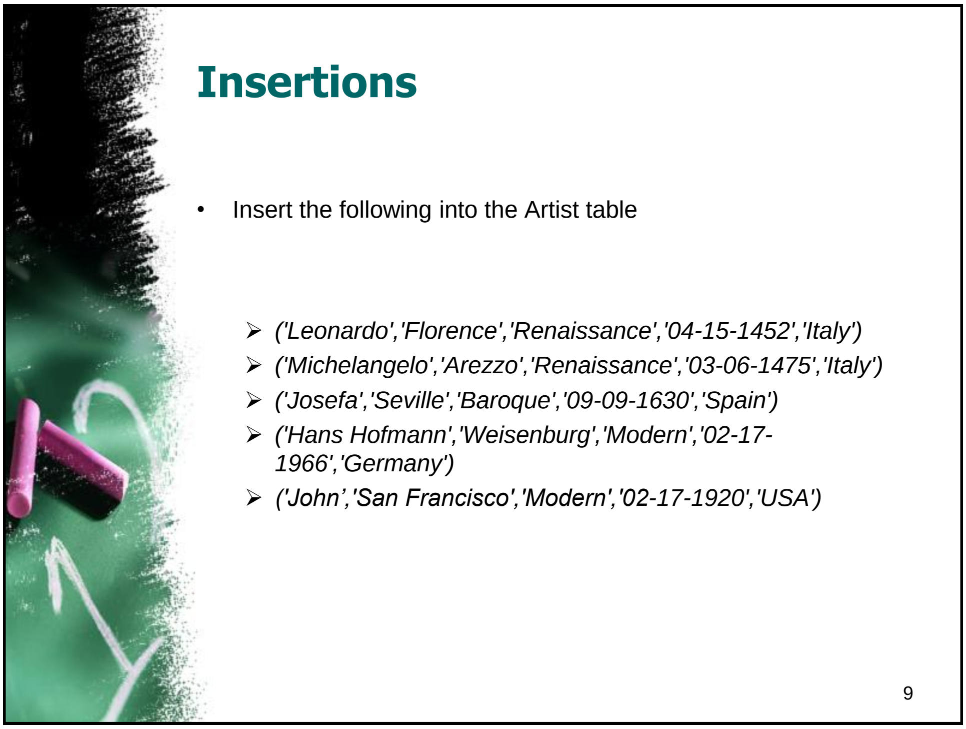
More on SELECT statements

SELECT [DISTINCT] select-list
FROM from-list
WHERE record-qualification
GROUP BY grouping-list
HAVING group-qualification



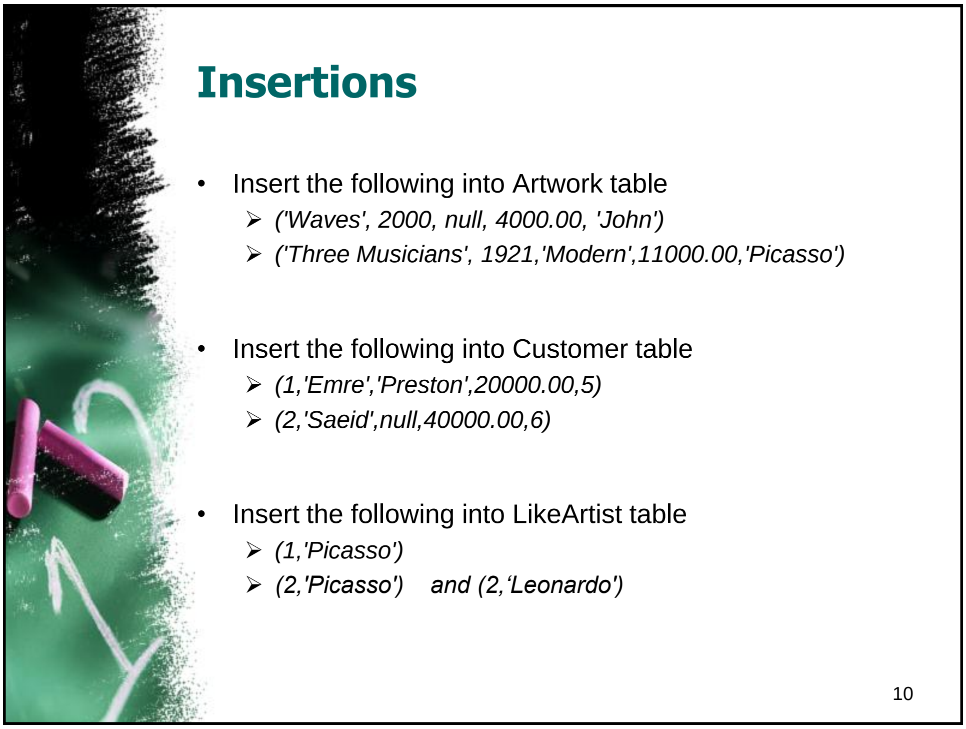
Your tasks

- You will insert more data into the Art database we just created last week.
- You will delete rows from a table
- Then, you’ll code queries involving single and multiple tables.



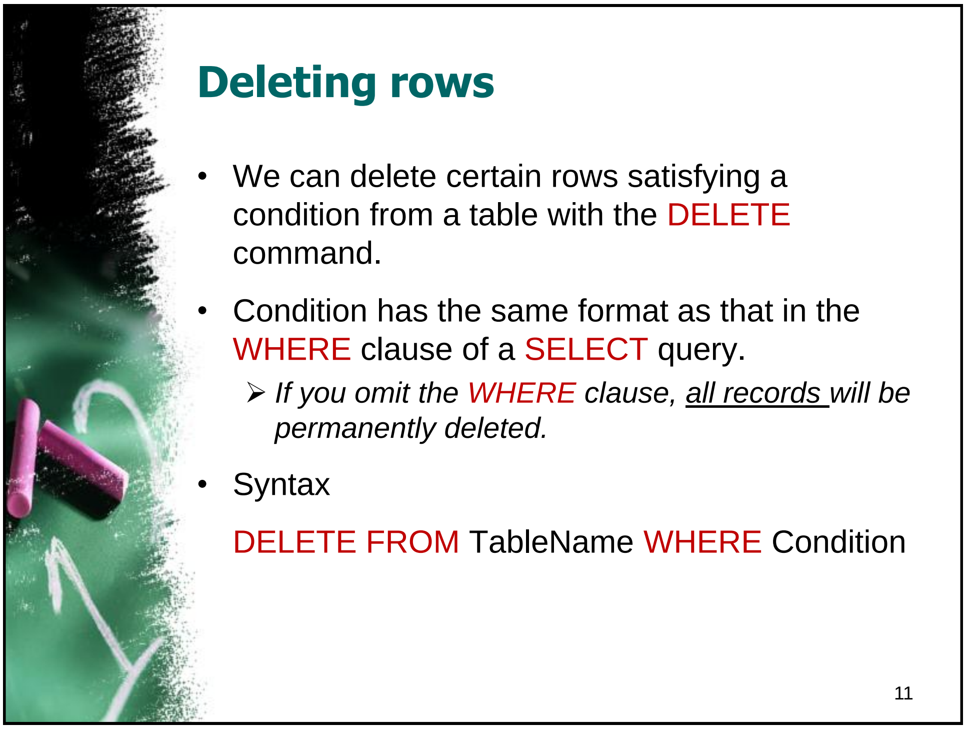
Insertions

- Insert the following into the Artist table
  - *('Leonardo','Florence','Renaissance','04-15-1452','Italy')*
  - *('Michelangelo','Arezzo','Renaissance','03-06-1475','Italy')*
  - *('Josefa','Seville','Baroque','09-09-1630','Spain')*
  - *('Hans Hofmann','Weisenburg','Modern','02-17-1966','Germany')*
  - *('John’,'San Francisco','Modern','02-17-1920','USA')*



Insertions

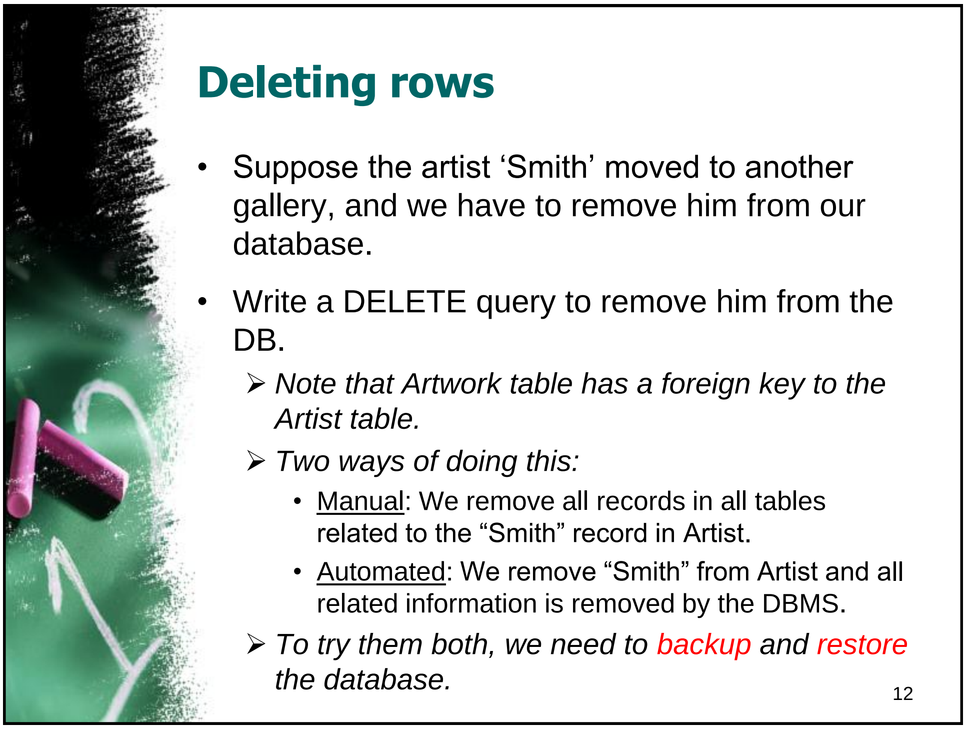
- Insert the following into Artwork table
  - *('Waves', 2000, null, 4000.00, 'John')*
  - *('Three Musicians', 1921,'Modern',11000.00,'Picasso')*
- Insert the following into Customer table
  - *(1,'Emre','Preston',20000.00,5)*
  - *(2,'Saeid',null,40000.00,6)*
- Insert the following into LikeArtist table
  - *(1,'Picasso')*
  - *(2,'Picasso')   and (2,‘Leonardo')*



Deleting rows

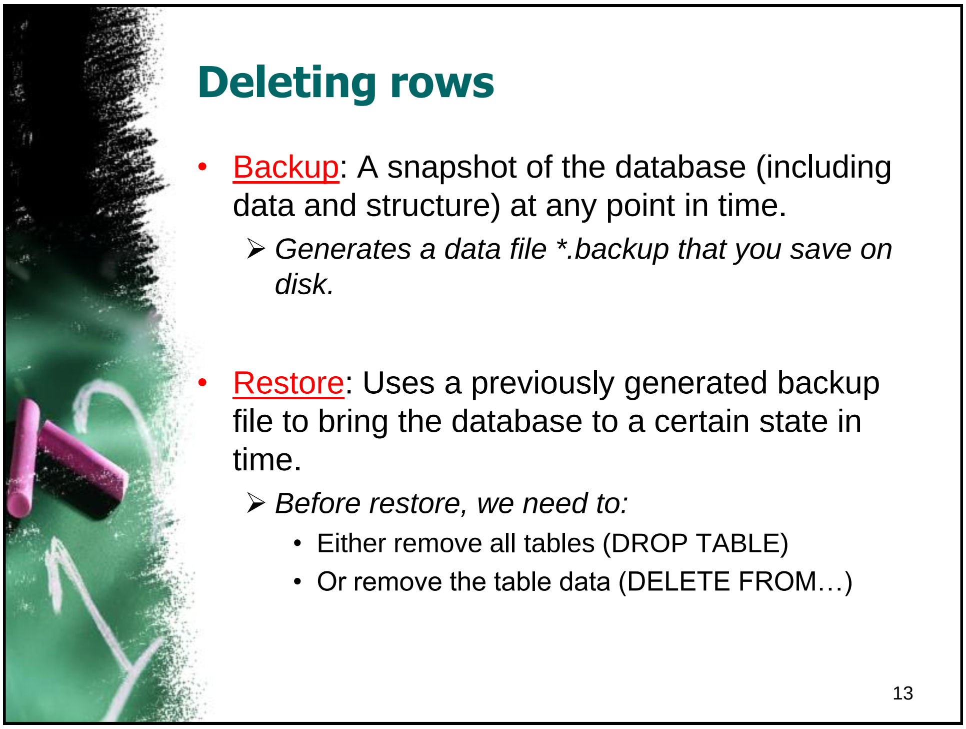
- We can delete certain rows satisfying a condition from a table with the DELETE command.
- Condition has the same format as that in the WHERE clause of a SELECT query.
  - *If you omit the WHERE clause, <u>all records</u> will be permanently deleted.*
- Syntax

  DELETE FROM TableName WHERE Condition



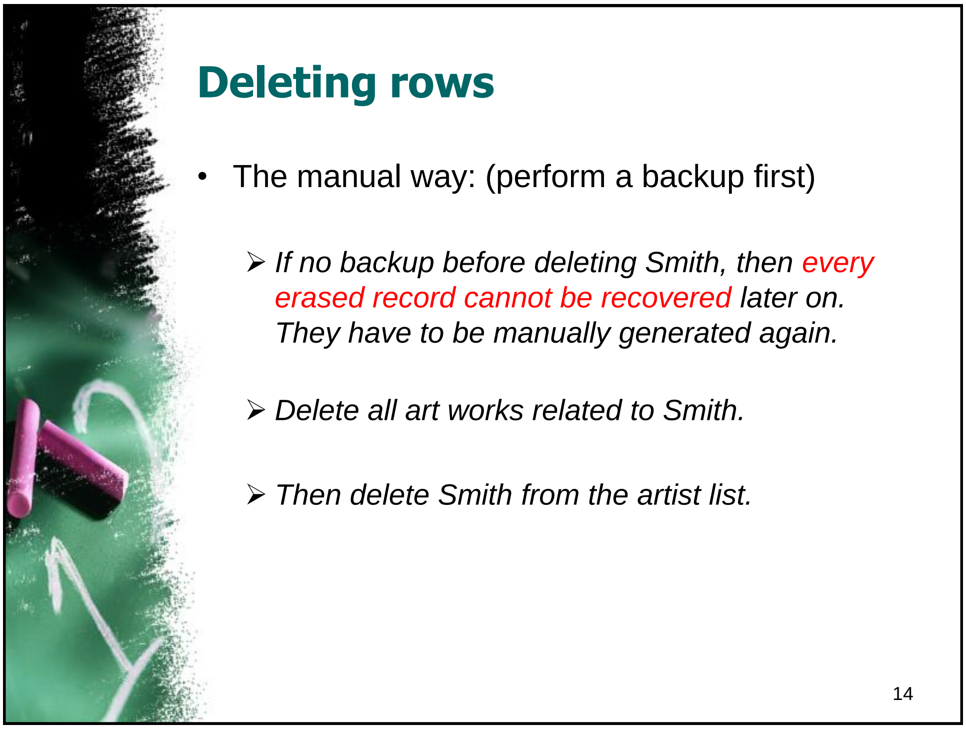
Deleting rows

- Suppose the artist ‘Smith’ moved to another gallery, and we have to remove him from our database.
- Write a DELETE query to remove him from the DB.
  - *Note that Artwork table has a foreign key to the Artist table.*
  - *Two ways of doing this:*
    - <u>Manual</u>: We remove all records in all tables related to the “Smith” record in Artist.
    - <u>Automated</u>: We remove “Smith” from Artist and all related information is removed by the DBMS.
  - *To try them both, we need to backup and restore the database.*



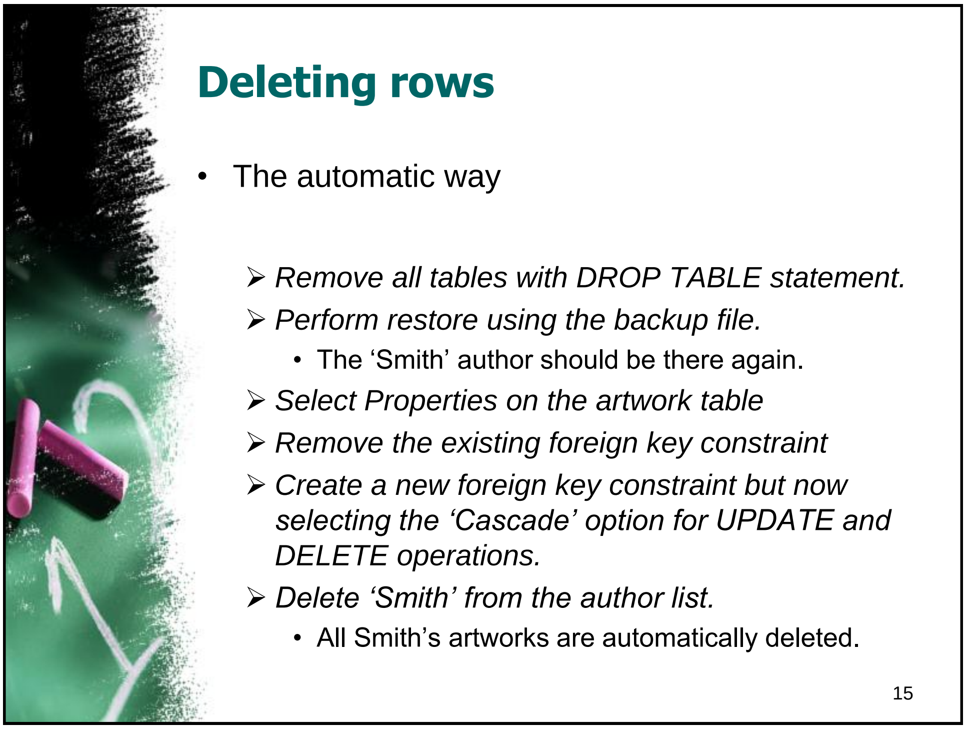
Deleting rows

- Backup: A snapshot of the database (including data and structure) at any point in time.
  - ➢ *Generates a data file *.backup that you save on disk.*
- Restore: Uses a previously generated backup file to bring the database to a certain state in time.
  - ➢ *Before restore, we need to:*
    - Either remove all tables (DROP TABLE)
    - Or remove the table data (DELETE FROM…)



Deleting rows

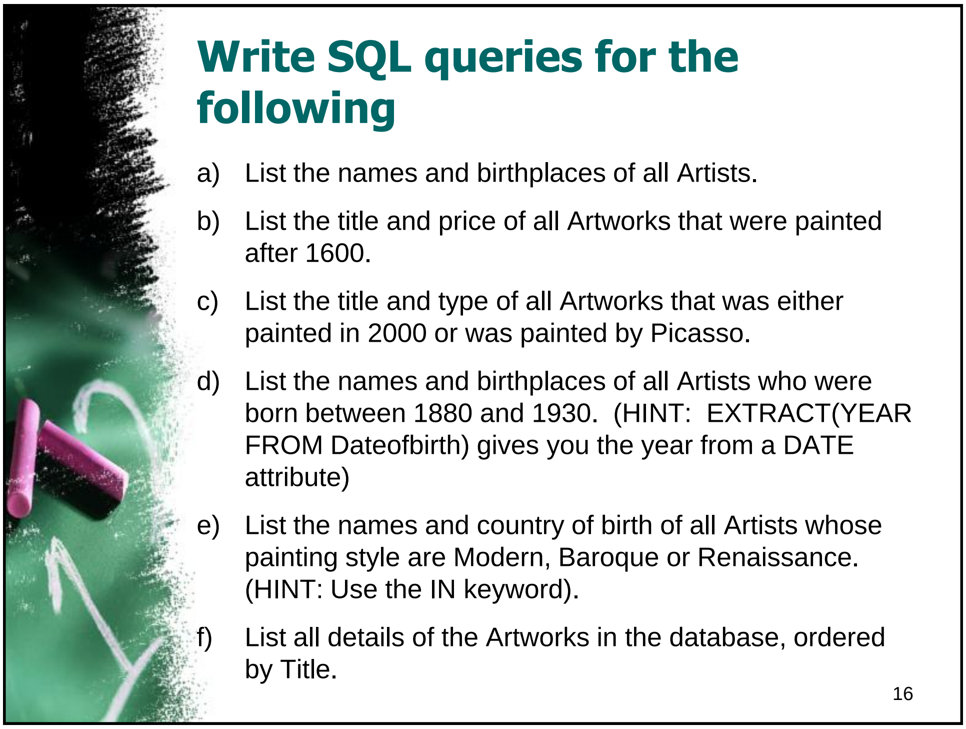
- The manual way: (perform a backup first)
  - ➢ *If no backup before deleting Smith, then every erased record cannot be recovered later on. They have to be manually generated again.*
  - ➢ *Delete all art works related to Smith.*
  - ➢ *Then delete Smith from the artist list.*



Deleting rows

- The automatic way

  - ➢ *Remove all tables with DROP TABLE statement.*
  - ➢ *Perform restore using the backup file.*
    - The ‘Smith’ author should be there again.
  - ➢ *Select Properties on the artwork table*
  - ➢ *Remove the existing foreign key constraint*
  - ➢ *Create a new foreign key constraint but now selecting the ‘Cascade’ option for UPDATE and DELETE operations.*
  - ➢ *Delete ‘Smith’ from the author list.*
    - All Smith’s artworks are automatically deleted.



Write SQL queries for the following

a) List the names and birthplaces of all Artists.

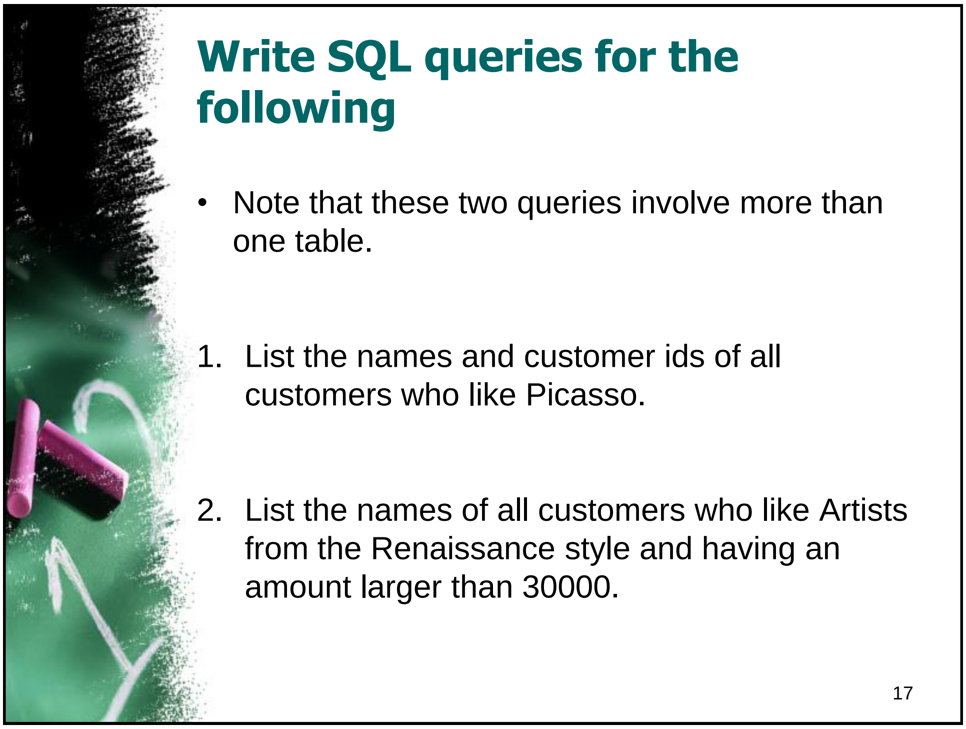
b) List the title and price of all Artworks that were painted after 1600.

c) List the title and type of all Artworks that was either painted in 2000 or was painted by Picasso.

d) List the names and birthplaces of all Artists who were born between 1880 and 1930. (HINT: EXTRACT(YEAR FROM Dateofbirth) gives you the year from a DATE attribute)

e) List the names and country of birth of all Artists whose painting style are Modern, Baroque or Renaissance. (HINT: Use the IN keyword).

f) List all details of the Artworks in the database, ordered by Title.

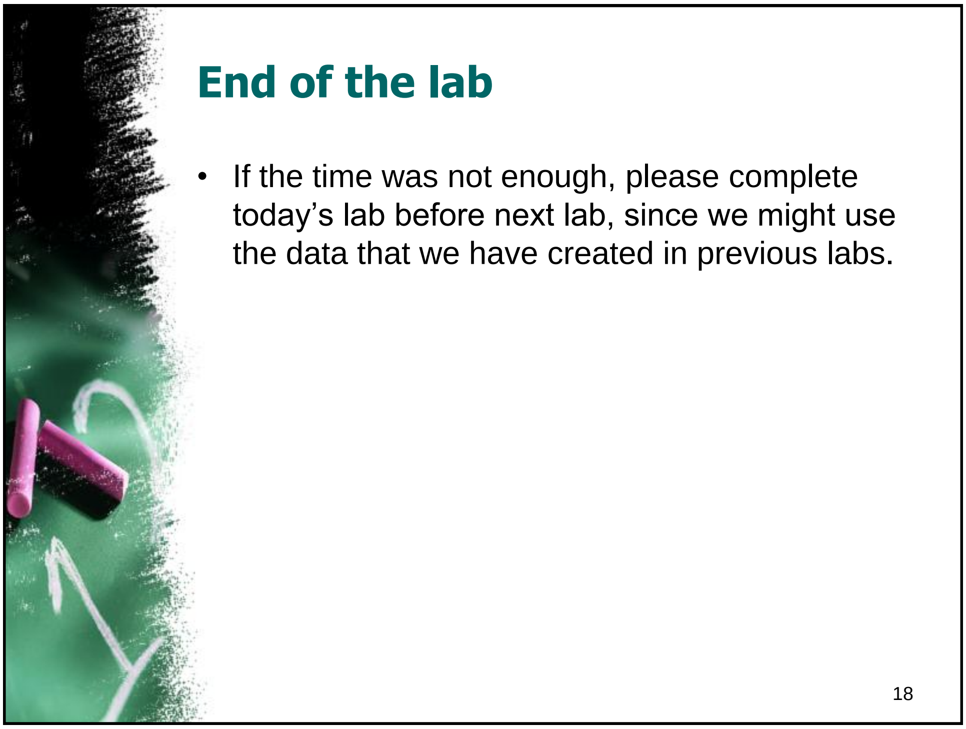


Write SQL queries for the following

- Note that these two queries involve more than one table.

1. List the names and customer ids of all customers who like Picasso.

2. List the names of all customers who like Artists from the Renaissance style and having an amount larger than 30000.



End of the lab

- If the time was not enough, please complete today’s lab before next lab, since we might use the data that we have created in previous labs.