

Introduction to Oracle SQL

Sample manual - first two chapters



TABLE OF CONTENTS (1 of 5)

1	DESIGNING DATABASES	Page
1.1	The Four Stages of Database Design	7
	<i>Stage 1 – Deciding what to Include</i>	7
	<i>Stage 2 – Dividing Data into Tables</i>	8
	<i>Stage 3 – Choosing a Primary Key for each Table</i>	9
	<i>Stage 4 – Creating Relationships and a Database Diagram</i>	10
1.2	Many-to-Many Relationships	11

2	SETTING UP ORACLE	Page
2.1	Installing Oracle Server	12
2.2	Installing Oracle SQL Developer	13
2.3	Creating a First Connection	14
	<i>What you Start With</i>	14
	<i>Creating a Connection to the System User</i>	15
2.4	Creating a Connection	16
	<i>Creating the User (and Schema)</i>	16
	<i>Creating a Connection for the User</i>	17
2.5	Installing and Using the Database	18

3	CREATING TABLES	Page
3.1	Creating a User and Schema	19
3.2	Creating a Table	20
	<i>Saving and Editing Changes</i>	20
	<i>Setting an Identity Column</i>	21
	<i>Setting Default Values</i>	21
3.3	Data Types	22
	<i>Text Data</i>	22
	<i>The Number Data Type</i>	23
	<i>Other Number Types</i>	23
	<i>Boolean Field Type</i>	24
	<i>Date/Time Data Types</i>	24
3.4	Entering Data in Tables	25

4	RELATIONSHIPS BETWEEN TABLES	Page
4.1	Understanding Relationships	26
4.2	Creating a Relationship	27
4.3	Creating Data Models	28
	<i>Using Model Browser to Create a New Design</i>	28
	<i>Adding Tables</i>	29
4.4	Formatting the Data Model	30
	<i>Simplifying what you See</i>	30
	<i>Improving the Lines</i>	30
	<i>Selecting Tables</i>	31
	<i>Changing Colours</i>	31
	<i>Using the Navigator</i>	32
4.5	Browser Operations	33
	<i>Saving Models</i>	33
	<i>Redisplaying Models</i>	34
4.6	Creating Subviews	35
4.7	Synchronising Models	36

TABLE OF CONTENTS (2 of 5)

5	CREATING QUERIES	Page
5.1	Working with Connections	37
	<i>Viewing Schemas</i>	37
	<i>Viewing Tables and Columns</i>	37
5.2	Setting Preferences	38
	<i>Change 1: Setting the Default Font Size</i>	38
	<i>Change 2: Changing the New Query Short-Cut</i>	38
	<i>Change 3: Widening the Scope of Completion Insight</i>	39
5.3	SQL Worksheets and Schemas	40
	<i>Creating SQL Worksheets</i>	40
	<i>Creating Another Worksheet Linked to the Same Schema</i>	41
	<i>Viewing and Changing a Worksheet's Schema</i>	41
	<i>Managing Query Windows</i>	41
5.4	Basic SELECT Statements	42
	<i>Where to Put your Commas</i>	42
	<i>Using Completion Insight</i>	43
5.5	Running Queries	44
	<i>Running a Single Command</i>	44
	<i>Running All of a Worksheet's Commands</i>	44
	<i>Running All a Worksheet's Commands in Sequence</i>	45
	<i>Closing your Outputs</i>	45
5.6	Dealing with Errors	46
5.7	Saving and Retrieving Queries	47
	<i>Saving Queries</i>	47
	<i>Opening Queries</i>	47
5.8	Formatting Worksheet Queries	48
	<i>Using Upper and Lower Case</i>	48
	<i>Changing the Default Case of SQL Keywords</i>	48
	<i>Changing Indentation</i>	49
	<i>Auto-formatting SQL</i>	49
	<i>Colours in SQL</i>	50
5.9	Comments	51
	<i>Commenting Out Blocks of Code</i>	51

6	THE SELECT STATEMENT	Page
6.1	SELECT Statement Syntax	52
	<i>Mnemonic for Order of Commands</i>	52
6.2	Table Aliases	53
	<i>Reason 1 for Aliases – Easier to Refer to Field Names</i>	53
	<i>Reason 2 for Aliases – Joins</i>	54
	<i>Changing a Table Alias</i>	54
6.3	Column Aliases	55
	<i>Basic Column Aliases</i>	55
	<i>The Word AS is Optional</i>	55
	<i>Aliases in WHERE and ORDER BY Clauses</i>	56
6.4	Ordering Rows	57
	<i>Simple Sorting</i>	57
	<i>Sorting by Multiple Columns</i>	57
	<i>Sorting Nulls</i>	58
6.5	Miscellaneous SELECT Tricks	59
	<i>Selecting All Columns Using the Asterisk</i>	59
	<i>Selecting Unique Rows</i>	59
	<i>Showing Top and Bottom Rows</i>	60
	<i>Including Ties</i>	60
6.6	Using UNION to Combine Results	61
6.7	Combining Data from Different Schemas	62

TABLE OF CONTENTS (3 of 5)

7	CRITERIA USING WHERE	Page
7.1	The WHERE Clause	63
	<i>Relational Operators</i>	63
7.2	Criteria with Numbers	64
	<i>Using Comparisons</i>	64
	<i>Finding Numbers in a Given Range</i>	64
7.3	Criteria using Text	65
	<i>Exact Matches</i>	65
	<i>Wildcard Matches using LIKE</i>	65
	<i>Making Searches Case Insensitive</i>	66
	<i>Using Special Characters with LIKE</i>	66
	<i>Ranges and Regular Expressions</i>	67
7.4	Criteria for Dates	68
	<i>Using Dates in Criteria</i>	68
7.5	Combining Criteria	69
	<i>Example using AND</i>	69
	<i>Example using OR</i>	69
	<i>Criteria Explained using Venn Diagrams</i>	70
	<i>Using Parentheses</i>	70
7.6	Filtering for Nulls	71
	<i>An Example of Testing for Nulls</i>	71

8	IMPORTING AND EXPORTING DATA	Page
8.1	Copying and Pasting	72
8.2	Exporting Query Results to Excel	73
8.3	Exporting Table Data	74
	<i>Step 1 – Starting the Export</i>	74
	<i>Step 2 – Choosing Columns</i>	75
	<i>Step 3 – Viewing the Resulting File</i>	75
8.4	Loading Data in Power Query	76

9	CALCULATIONS	Page
9.1	Creating Calculated Columns	77
	<i>Giving Calculated Columns Aliases</i>	77
	<i>Using Column Aliases in ORDER BY Clauses</i>	77
	<i>Column Aliases Don't Work in WHERE Criteria</i>	78
9.2	Using SQL Functions	79
	<i>Typing an SQL Function</i>	79
	<i>Getting Help on Oracle SQL Functions</i>	79
9.3	Casting Data Types	80
	<i>What is Casting?</i>	80
	<i>Casting Using the CAST Function</i>	80
	<i>Converting Text to Dates</i>	81
	<i>Converting Text to Numbers</i>	81
9.4	Numerical Calculations	82
	<i>Mathematical Symbols and BODMAS</i>	82
	<i>The Modulus Function (MOD)</i>	82
	<i>Powers and Roots</i>	83
	<i>Rounding</i>	83
	<i>Other Useful Mathematical Functions</i>	83
9.5	Text Calculations	84
	<i>Concatenating Text</i>	84
	<i>Changing the Case of Text</i>	84
	<i>Getting Numerical Information</i>	85
	<i>Manipulating Text</i>	86
9.6	Examples of Text Functions	87
	<i>Example: Counting Words</i>	87
	<i>Example: Getting the Last Word</i>	87
9.7	Dealing with Nulls	88
	<i>The COALESCE Function</i>	88
	<i>The NVL Function</i>	89

10	THE CASE EXPRESSION	Page
10.1	The Searched Case Expression	90
	<i>Example: Film Bands</i>	90
	<i>Example: Film Era</i>	91
	<i>Using CASE Expressions in WHERE Criteria</i>	92
10.2	The Simple Case Statement	93
10.3	Nested CASE Statements	94

TABLE OF CONTENTS (4 of 5)

11	DATE CALCULATIONS	Page	12	JOINS	Page
11.1	How Dates and Times Work	95	12.1	Overview of Joins	107
	<i>Using Dump to See Dates</i>	95		<i>What is a Join?</i>	107
	<i>The Seven Parts of a Date/Time</i>	95		<i>The Types of Join</i>	107
11.2	Basic Dates and Times	96	12.2	Understanding your Database	108
	<i>Date and Time Types</i>	96		<i>How Relationships Work (Reminder)</i>	108
	<i>Getting the Current Date/Time</i>	96	12.3	Inner Joins	109
11.3	Representing Dates (and Times)	97		<i>The Syntax of an Inner Join</i>	109
	<i>The TO_CHAR Function</i>	97		<i>Our Example – Joining the Film and Director Tables</i>	109
	<i>The Main Components of a Date</i>	98		<i>Joining more than One Table</i>	110
	<i>The FM Modifier</i>	99		<i>Variations on Inner Join Syntax</i>	111
	<i>Ordinal Suffices</i>	99		<i>Composite Joins</i>	112
	<i>Putting Everything Together</i>	99		<i>Joining by Expressions</i>	112
	<i>Time Format Codes</i>	100	12.4	Outer Joins	114
11.4	Extracting Parts of Dates	101		<i>Creating Outer Joins in SQL</i>	114
	<i>The Extract Function</i>	101		<i>Right Outer Joins</i>	114
	<i>Keywords for Extracting Dates</i>	102		<i>Picking Out Unmatched Rows</i>	115
	<i>Keywords for Extracting Times</i>	102		<i>Full Outer Joins</i>	115
	<i>Truncating Dates/Times</i>	103	12.5	Cross Joins	116
	<i>Rounding Dates</i>	103		<i>A Practical Example of Cross Joins</i>	116
11.5	Getting the Difference Between Dates	104	12.6	Self-Joins	117
	<i>Simple Date Subtraction</i>	104			
	<i>Date/Time Subtraction Returning Intervals</i>	104			
	<i>Getting True Ages using MONTHS_BETWEEN</i>	105			
11.6	Adding Dates	106			
	<i>Adding and Subtracting Days</i>	106			
	<i>Adding and Subtracting Other Time Intervals</i>	106			

TABLE OF CONTENTS (5 of 5)

13	SUMMARISING DATA	Page
13.1	Simple Summarising	118
	<i>Syntax of a Simple Summary</i>	118
13.2	Counting	119
	<i>Counting All of a Table's Rows</i>	119
	<i>Counting Non-Null Columns</i>	119
	<i>Counting CLOB fields</i>	119
	<i>Counting Unique Values</i>	120
	<i>Using Filter to Count a Subset of Rows</i>	120
13.3	Grouping	121
	<i>Why you Need GROUP BY</i>	121
	<i>The GROUP BY Clause</i>	121
	<i>Grouping by Multiple Columns</i>	122
	<i>Grouping by Expressions</i>	122
	<i>Grouping without Aggregating</i>	123
13.4	Filtering Results using HAVING	124
13.5	Dealing with Nulls	125
	<i>The Default Treatment of Nulls</i>	125
	<i>Forcing SQL to Include Nulls</i>	125
13.6	The CUBE and ROLLUP Operators	126
	<i>Using CUBE to Show All Combinations</i>	126
	<i>Using ROLLUP to Show Most Combinations</i>	127
13.7	String Aggregation	128

14	VIEWS	Page
14.1	What Views Are	129
14.2	Why Views are Useful	130
	<i>Use 1: Pre-Joining Tables</i>	130
	<i>Use 2: Virtually Renaming Columns</i>	131
14.3	Creating Views	131
14.4	Changing Views	133
	<i>Method 1 - Scripting Changes</i>	133
	<i>Method 2 – Using the Edit View Window</i>	134
	<i>Method 3 – Typing your Instructions</i>	134
14.5	Sorting in Views	135
14.6	Joining to Views	136

15	CTES AND DERIVED TABLES	Page
15.1	Multi-Stage Queries	137
15.2	Derived Tables	138
	<i>Single Derived Tables</i>	138
	<i>Multiple Derived Tables</i>	139
15.3	Single CTEs (Common Table Expressions)	140
	<i>Syntax of Single CTEs</i>	140
	<i>The CTE for our Example</i>	140
15.4	Multiple CTEs	141
	<i>Syntax of Multiple CTEs</i>	141
	<i>Example of a Multiple CTE</i>	142

16	SUBQUERIES	Page
16.1	Creating Subqueries	143
	<i>Example: Showing the Name of the Longest Film</i>	143
	<i>ANY, ALL, IN and NOT IN</i>	144
	<i>Using EXISTS to Check whether Rows are Returned</i>	145
16.2	Correlated Subqueries	146
	<i>Definition and Example</i>	146
	<i>Alternatives to Correlated Subqueries</i>	146

17	ORACLE SQL AND AI	Page
17.1	Choosing an AI Tool	147
17.2	Blind Queries	148
17.3	Querying a Specific Database	149
	<i>Creating a Backup of a Schema's Tables</i>	149
	<i>Uploading your Schema File</i>	150
	<i>Requesting a Query</i>	150
17.4	Tidying Up Queries	151
17.5	Optimising Queries	152
17.6	Debugging SQL	153

CHAPTER 1 - DESIGNING DATABASES

The world runs on relational databases. If you understand the principles upon which these are built, you'll find it much easier to write SQL to get information out of them!



*This manual gives an overview only of database design principles. If you want to delve deeper, try Googling phrases like **Third Normal Form**, **Database Normalisation** or **Entity Diagram**. If nothing else, this will give you an impressive search history in your browser!*

1.1 The Four Stages of Database Design

There are four stages to designing a relational database, shown below (using the example of creating a simple database to hold films; or movies, if you must).

Stage 1 – Deciding what to Include

A good way to do this is to create a spreadsheet of the data you want to include for each film:

Title	Oscars	Director	Date of birth	Studio
Armageddon	0	Michael Bay	17/02/1965	Touchstone Pictures
Bad Boys	0	Michael Bay	17/02/1965	Jerry Bruckheimer Films
Bad Boys II	0	Michael Bay	17/02/1965	Jerry Bruckheimer Films
Dead Poets Society	1	Peter Weir	21/08/1944	Touchstone Pictures
Master and Commander ...	2	Peter Weir	21/08/1944	20th Century Fox
Pearl Harbor	1	Michael Bay	17/02/1965	Touchstone Pictures
The Rock	0	Michael Bay	17/02/1965	Hollywood Pictures
The Truman Show	0	Peter Weir	21/08/1944	Scott Rudin Productions

We want to assign each film to a director, but we don't want to have to type each director's name in over and over again!

The aim of designing a relational database is to ensure that you don't hold information twice:

Title	Oscars	Director
Dead Poets Society	1	Peter Weir
Master and Commander ..	2	Peter Wier
The Truman Show	0	Peter Weird

Not only is holding duplicate information inefficient, but it also means that spelling mistakes will creep in. Here listing out films directed by **Peter Weir** would miss out the last two films, as his name has been misspelled.

Stage 2 – Dividing Data into Tables

Having decided what data you want to include, the next stage of database design is to decide which table each bit of information belongs to:

Title	Oscars	Director	Date of birth	Studio
Armageddon	0	Michael Bay	17/02/1965	Touchstone Pictures
Bad Boys	0	Michael Bay	17/02/1965	Jerry Bruckheimer Films
Bad Boys II	0	Michael Bay	17/02/1965	Jerry Bruckheimer Films
Dead Poets Society	1	Peter Weir	21/08/1944	Touchstone Pictures
Master and Commander ...	2	Peter Weir	21/08/1944	20th Century Fox
Pearl Harbor	1	Michael Bay	17/02/1965	Touchstone Pictures
The Rock	0	Michael Bay	17/02/1965	Hollywood Pictures
The Truman Show	0	Peter Weir	21/08/1944	Scott Rudin Productions

↑

These are all details to do with the film itself.

↑

These are to do with the director (name / birthday).

↑

These are details to do with the studio.



There's no magic wand to make this easier, other than bitter experience of getting it wrong and having to start again! A good guideline is that if you find yourself typing in something twice, it probably belonged in a different table.

For our example above, there are clearly 3 separate entities: films, the directors who made them and the studios which produced them. Here are the fields that each table could contain:

Table	Fields
<i>Film</i>	Title and Oscars Won , plus something to identify which director and which studio made it
<i>Director</i>	Director name and Date of birth , plus some unique identifier for the director
<i>Studio</i>	Studio name , plus some unique identifier for the studio

What you now need to do next is to decide what form these unique identifiers should take.

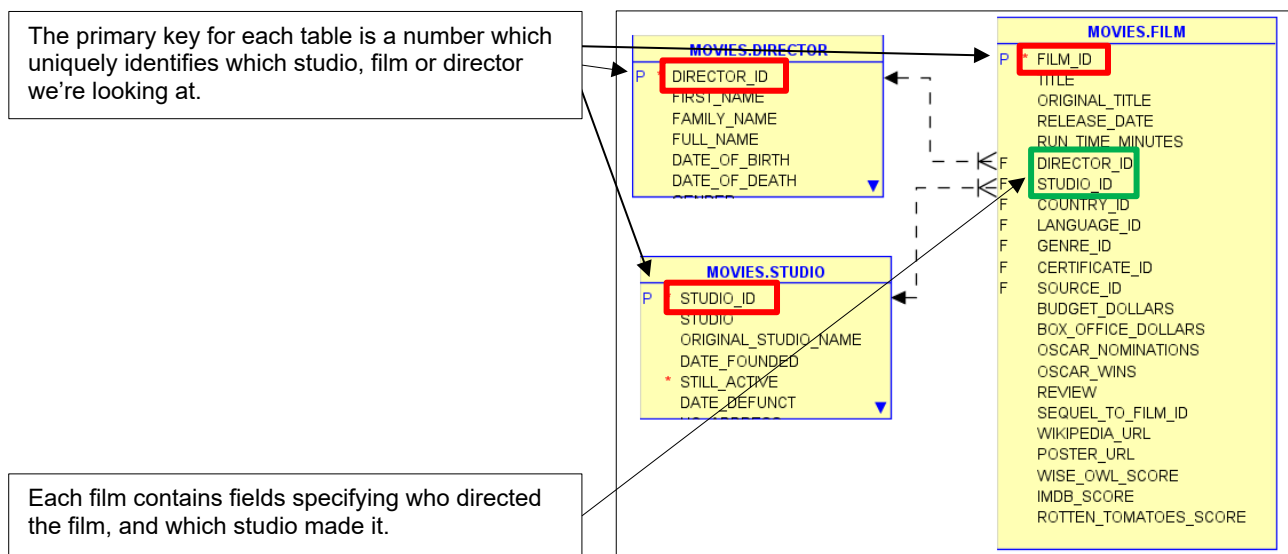
Stage 3 – Choosing a Primary Key for each Table

The *primary key* for a table is a field which tells you exactly which record you're considering (for example, if you know a film's **director_id** you can look up all of the director's other details).



From the above definition, it follows that two records in a table can't have the same value for the primary key field – the field is *unique*.

For our example, we could use the director and studio names as our primary keys, but Oracle works most efficiently if the primary key is as short as possible, so we'll create new fields instead:



Here's what **The Sound of Music** would now look like:

FILM_ID	TITLE	OSCAR_WINS	DIRECTOR_ID	STUDIO_ID
1	638 The Sound of Music	5	89	4

Including the director's unique number allows us to look up all their other details:

DIRECTOR_ID	FULL_NAME	DATE_OF_BIRTH	GENDER
89	Robert Wise	10-SEP-14	Male

Including the studio's unique number allows us to look up its name:

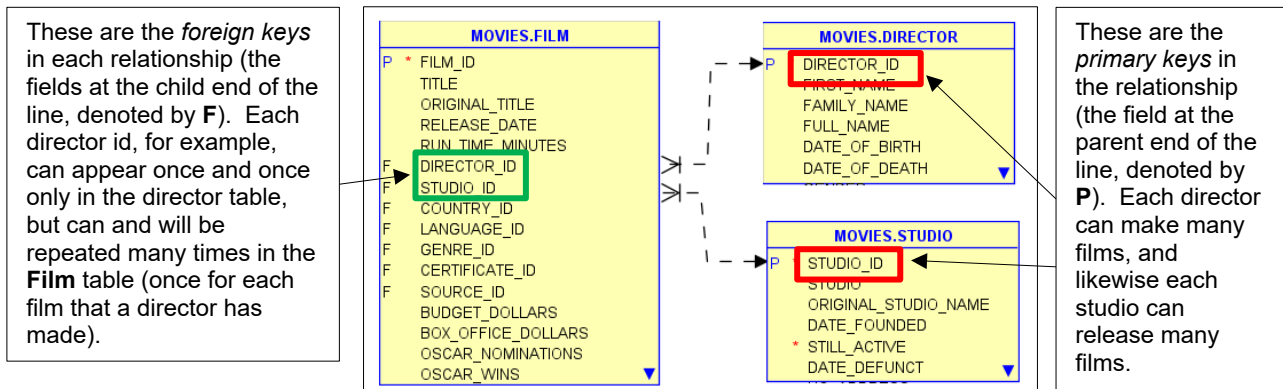
STUDIO_ID	STUDIO
4	20th Century Studios



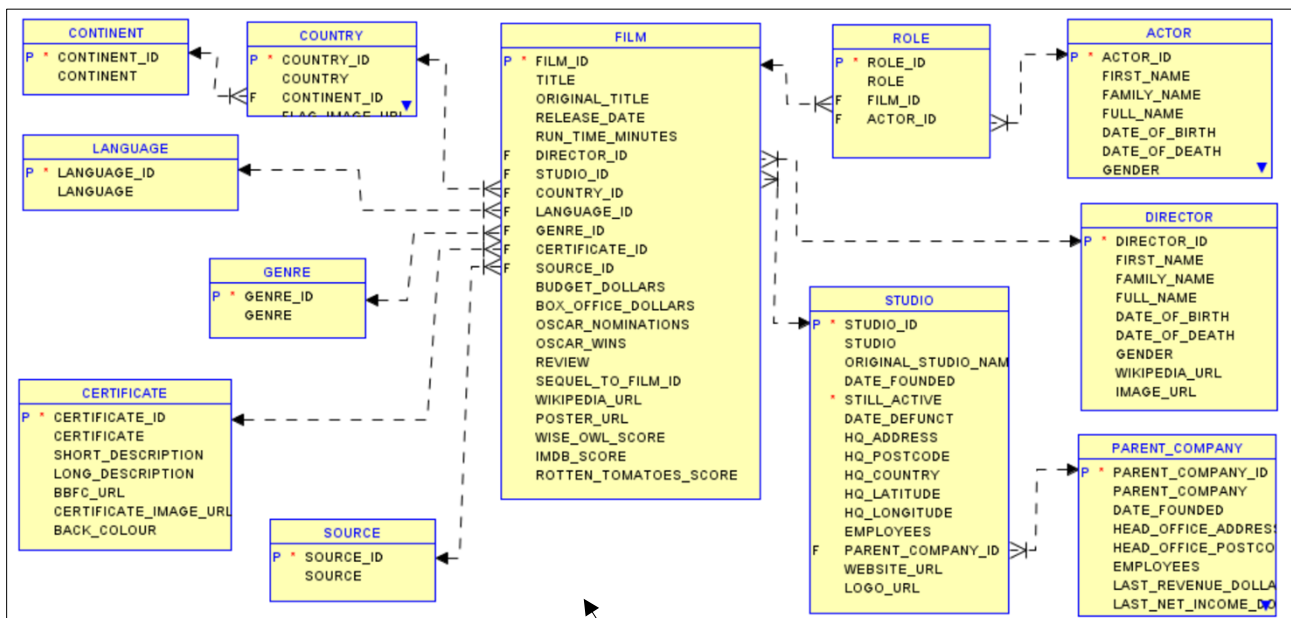
If you're beginning to think that relational databases are just like a lot of spreadsheets joined together with a more efficient version of a **VLOOKUP** or **XLOOKUP** formula in Excel, you're absolutely right!

Stage 4 – Creating Relationships and a Database Diagram

The last step in designing a database is to decide for each relationship that you create whether it is *one-to-many* or *many-to-one* (parent-child or child-parent):



Database diagrams often involve hundreds of tables:



Our **Movies** database contains just 12 tables, all of which are shown above, and hence is untypically simple.

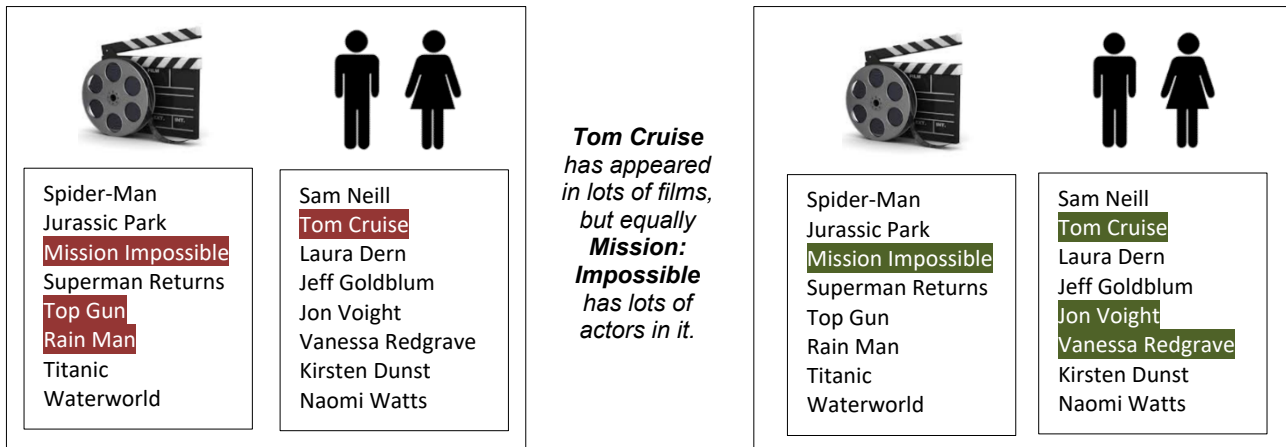


**Wise
Owl's
Hint**

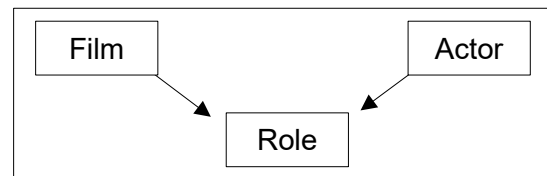
The above diagrams were created using Data Modeler within SQL Developer (a later chapter in this courseware shows how to create diagrams like this).

1.2 Many-to-Many Relationships

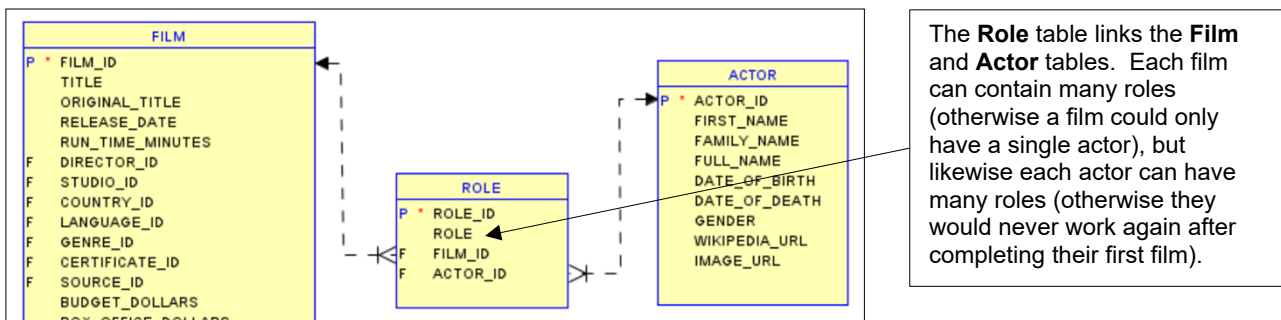
There's no such thing as a many-to-many relationship in Oracle, but they do exist in real life:



The solution to this problem is to create a table that is a child to both of the two parent tables, as here:



Here's what the database would look like:



Here are 3 rows from the **Role** table:

Film number 1 appears twice in this list, as does actor number 1.

ROLE_ID	ROLE	FILM_ID	ACTOR_ID
1	1 Ray Ferrier	33	1
2	2 Dr. Alan Grant	1	2
3	3 Dr. Ellie Sattler	1	3
4	202 Nathan Algren	41	1

Here are the films and actors who are represented by these rows of data (the duplicate film name was **Jurassic Park**, and duplicated actor turns out to be **Tom Cruise**).

ROLE_ID	ROLE	FILM_ID	ACTOR_ID	TITLE	FULL_NAME
1	1 Ray Ferrier	33	1	1 War of the Worlds	Tom Cruise
2	2 Dr. Alan Grant	1	2	2 Jurassic Park	Sam Neill
3	3 Dr. Ellie Sattler	1	3	3 Jurassic Park	Laura Dern
4	202 Nathan Algren	41	1	1 The Last Samurai	Tom Cruise

CHAPTER 2 - SETTING UP ORACLE

2.1 Installing Oracle Server

You are unlikely to need this (an Oracle database will surely already have been set up on your network by your IT people), but just in case:

- a) Go to the Oracle website <https://www.oracle.com/database/free/> and click to download Oracle for Windows.

Oracle AI Database Free for Windows

Develop with Oracle AI Database 26ai for Windows x64. Download from the link below.

[Download for Windows](#)

[Windows \(64-bit\)](#)

oracle-ai-database-free-26ai-23.26.2.windows.x64.zip
1,385,487,312 bytes
SHA256 bdc860145a443230bd4f94553ce2e8da1e93239

- [Oracle AI Database Free Installation Guide](#)
- [Oracle AI Database Client Installation Guide](#)
- [Installation and Getting Started Video](#)

FREEInstall.rsp

Oracle AI Database 26ai Free.msi

setup.exe

Setup.ini

- b) After extracting the installation files, click on the setup executable to install Oracle.

This password will be used for SYS, SYSTEM and PDBADMIN accounts.

Enter Database Password

Confirm Database Password

- c) Be sure to make a note of the password you set, as you'll need it to set up users and databases.

Be warned that the installation takes about half an hour even on a fast laptop! When you've finished, check that your Oracle services are running:

Run

Type the name of a program, folder, resource, and Windows will open

Open:

☐ Run in separate memory space

OK Cancel

Hold down the Windows key and press **R** to show this dialog box, then go to **services.msc** and check your Oracle services are running automatically:

Service Name	Status	Startup Type	Log On As
OracleJobSchedulerFREE	Stopped	Manual	NT SERVICE...
OracleOraDB23Home1TNSListener	Running	Automatic	NT SERVICE...
OracleServiceFREE	Running	Automatic	NT SERVICE...
OracleVssWriterFREE	Running	Automatic	NT SERVICE...



If you're logged on using one of your organisation's domain accounts you'll need to be either in your office or connected using a VPN to get through an early security check. You'll also need to be a member of the **Administrators** group on your laptop.

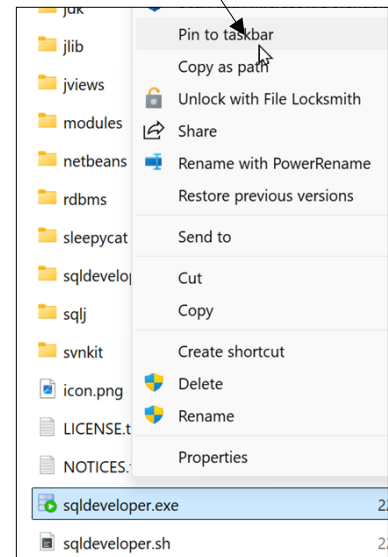
2.2 Installing Oracle SQL Developer

Most people use Oracle's SQL Developer to get at data in their databases. You don't actually install this software, but instead copy it onto your computer and run it:

- a) Go to the Oracle download site for SQL developer – at the time of writing: <https://www.oracle.com/database/sqldeveloper/technologies/download/>



- b) Extract the files to a directory, then right-click on **sqldeveloper.exe** to run it or pin it to your desktop or taskbar.



- c) Once SQL Developer has loaded you can use it to create and interrogate databases, as explained in the rest of this chapter.



If you're used to using Visual Studio for Code (VSC), you may prefer to install the Oracle SQL Developer for VSCode extension and create queries in VSC instead.

2.3 Creating a First Connection

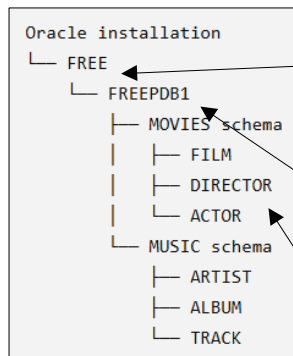
When you start with Oracle there are some things which will have been created for you, and other things you'll need to add in order to eventually set up the **Movies** and **Music** schemas.



If you come from a SQL Server background you should know that a SQL Server database is called a schema in Oracle, and typically all the schemas that you create will belong to a single PDB (Pluggable Database).

What you Start With

When you install Oracle, you get the following:



FREE is the *container database (CDB)* created. All schemas that you create will belong to *pluggable databases (PDBs)*, and these in turn will belong to the container database.

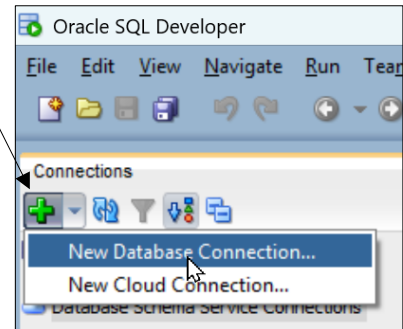
FREEPDB1 is a pluggable database created for you. There is no point creating any additional pluggable database as the **Movies** and **Music** schemas that we will create can both live in the default **FREEPDB1** pluggable database.

The **Movies** and **Music** schemas don't exist initially – it's up to you to create these, which we'll do later in this chapter.

Creating a Connection to the System User

By default you already have a **SYS** user with access to the top-level CDB, but you should only use this if you're a database admin. Instead, create another connection for your **FREEPDB1** PDB:

- a) Click on the green French pharmacy sign to add a connection to the **FREEPDB1** database that you've been given.

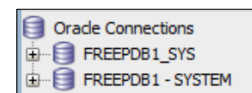


- b) It doesn't matter what you call your connection, but including the name of the database and the name of the user seems sensible.

- c) You are connecting as the built-in **SYSTEM** user. You did remember the password you set when you installed Oracle, didn't you?

- d) Click on **Service name** and type in the **FREEPDB1** database to which you want to connect.

- e) Click on the **Test** button to check your connection, then click on **Save** to save it. You should now have a new connection listed:



*If none of this makes much sense, just accept that you should avoid using the built-in **SYS** account for things to do with your top-level **FREEPDB1** database, so you're creating the ability to connect as **SYSTEM** instead.*

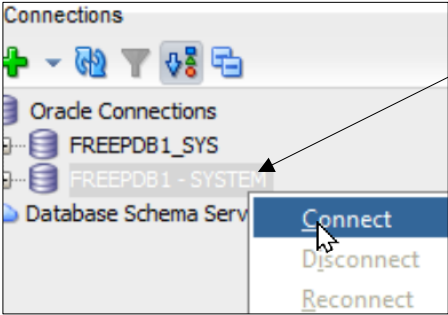
2.4 Creating a Connection

To connect to the **Movies** database referenced throughout this courseware you now need to create a user and a schema (and repeat the steps for the **Music** database used for the course exercises).

Creating the User (and Schema)

The first thing to do is to create your new user, and give them sufficient access rights to do things:

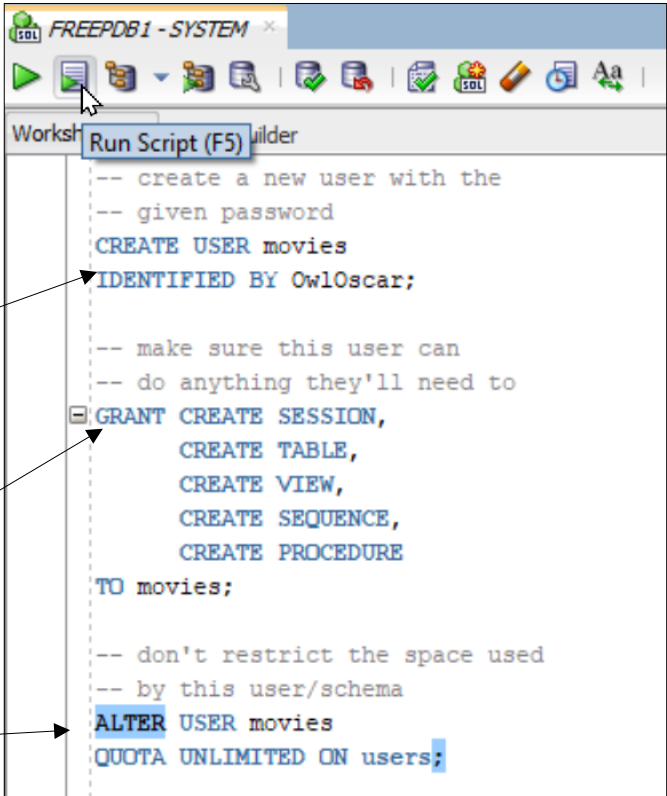
a) Right-click and choose to connect to the top-level database **FREEPDB1** as the all-powerful **SYSTEM** user.



b) Create a new user with the given name and password.

c) Give this user the right to create objects.

d) Make sure that you won't get space overflow messages when you try to create or populate tables for your user.



```
-- create a new user with the
-- given password
CREATE USER movies
IDENTIFIED BY OwlOscar;

-- make sure this user can
-- do anything they'll need to
GRANT CREATE SESSION,
CREATE TABLE,
CREATE VIEW,
CREATE SEQUENCE,
CREATE PROCEDURE
TO movies;

-- don't restrict the space used
-- by this user/schema
ALTER USER movies
QUOTA UNLIMITED ON users;
```

When you've finished typing this in press **F5** to run the 3 statements that you've created.

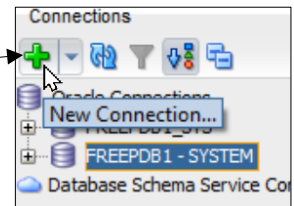


*When you create a user, you also automatically create a schema that the user owns (so the above example would create a schema called **movies** too).*

Creating a Connection for the User

You now need to create a connection for your **movies** user:

Start by clicking on the green cross to create a new connection.



You can then fill in the details for this connection:

a) Give this connection a name (this is what will appear in the list of connections).

b) The user name (as this shows) isn't case sensitive. The password for this example will be **OwIOscar**, since that's what you set when you created the user.

c) It's definitely worth ticking this box, so you don't keep having to type in a password when you connect to your database.

d) Choose to type in a service name, and store this connection within the main (and only) **FREEDB1** pluggable database.

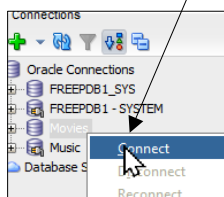
2.5 Installing and Using the Database

Your courseware should include two files with names like these:

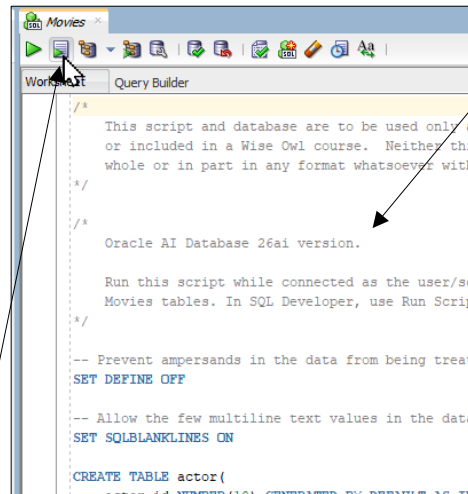
Database	File name
<i>Movies</i>	Create Oracle movies_02 database.sql
<i>Music</i>	Create Oracle music_01 database.sql

Here's how to install the **movies** database, for example:

- a) Start by right-clicking on the **Movies** connection and choosing to connect to it.



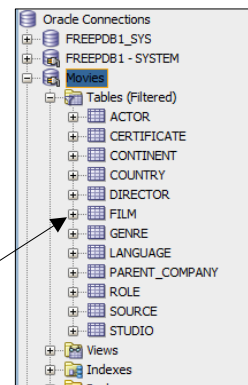
- c) Click on this icon or press **F5** to run all of the statements in your SQL.



- b) Paste in the SQL contained in the file shown above.

After a few minutes the SQL should finish running (make sure you check down looking for errors). You should now have a set of tables:

If all goes well you should have a list of tables when you expand the **Movies** database. Even though internally the tables were in lower case, Oracle will always display them in upper case (this is why you should use underscores and not CamelCase to separate the parts of a table name: so **parent_company** rather than **ParentCompany**, for example).



Our training and consultancy services

		Basic training	Advanced training	Systems / consultancy
Desktop	Microsoft Excel			
	VBA macros			
	Microsoft Access			
Business Intelligence	Power BI			
	Power Apps			
	Power Automate			
SQL Server	SQL			
	Reporting Services			
	Report Builder			
	Integration Services			
	Analysis Services			
Coding	Visual C# programming			
	VB programming			
	DAX			
	Python			



WiseOwl
Training