

CSCE 156 – SQL Supplemental Example Sheet

Query Type	Syntax Example	Notes
Simple SELECT query	<code>SELECT * FROM Albums;</code>	Returns all columns of all records in the Albums table.
Simple SELECT query with column aliases	<code>SELECT AlbumTitle AS Title, AlbumId AS Id FROM Albums;</code>	Selects only the two specified columns, aliased (renamed) as shown from all records in the albums database
Simple SELECT query with a WHERE clause	<code>SELECT AlbumTitle FROM Albums WHERE AlbumYear > 2000;</code>	Selects the title and album id of all albums created after the year 2000
Boolean operators	<code>SELECT AlbumTitle FROM Albums WHERE AlbumYear > 2000 AND AlbumNumber = 1;</code>	Selects the title of all albums which where a bands first release after the year 2000
LIKE operator and %	<code>SELECT AlbumTitle FROM Albums WHERE AlbumTitle LIKE '%fire%'</code>	Select all album titles which contain the word "fire"
Like operator and _	<code>SELECT BandName FROM Bands WHERE BandName LIKE '_2'</code>	Select all band names which have exactly 2 letters where the second letter is "2"
Simple SELECT query with ORDER BY	<code>SELECT AlbumTitle, AlbumYear FROM Albums ORDER BY AlbumTitle;</code>	Selects the title and creation year of all albums alphabetically ordered by title
COUNT function	<code>SELECT COUNT(*) FROM Albums</code>	Returns the number of albums in the database
COUNT function with WHERE clause	<code>SELECT COUNT(*) FROM Albums WHERE AlbumYear > 2000;</code>	Returns the number of albums created after the year 2000
AVG function with WHERE clause	<code>SELECT AVG(TrackLength) from AlbumSongs WHERE TrackNumber =1</code>	Returns the average length of the first track of all albums
Conceptual JOIN	<code>SELECT Bands.BandName, Albums.AlbumTitle FROM Bands JOIN Albums;</code>	Selects all combinations of album titles and band names
Simple JOIN	<code>SELECT Bands.BandName, Albums.AlbumTitle FROM Bands JOIN Albums ON Bands.BandID = Albums.AlbumTitle ;</code>	Selects band names and the bands associated album titles
Simple JOIN with table aliases	<code>SELECT B.BandName, A.AlbumTitle FROM Bands AS B JOIN Albums AS A ON B.BandID = A.AlbumTitle ;</code>	Selects band names and the bands associated album titles

Query Type	Syntax Example	Notes
Simple JOIN with table aliases	<pre>SELECT S.SongTitle, A.TrackNumber, A.TrackLength FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID =A.SongID;</pre>	Selects the title, length and track number of all songs which appear in an album
Simple JOIN with table aliases with WHERE clause	<pre>SELECT S.SongTitle, A.TrackNumber, A.TrackLength FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID =A.SongID WHERE A.TrackLength < 60;</pre>	Selects the title, length and track number of all songs which appear in an album that are shorter than 1 minute
Simple JOIN with table aliases with WHERE and ORDER BY clauses	<pre>SELECT S.SongTitle, A.TrackNumber, A.TrackLength FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID =A.SongID WHERE A.TrackLength < 60 ORDER BY S.SongTitle;</pre>	Selects the title, length and track number of all songs which appear in an album that are shorter than 1 minute sorted alphabetically by title
Multiple JOINS with WHERE clause	<pre>SELECT S.SongTitle, A.TrackNumber, A.TrackLength FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID =A.SongID JOIN Albums as T ON A.AlbumID = T.AlbumID WHERE T.AlbumTitle = "Nevermind";</pre>	Selects the title, length and track number of all songs which appear in the album "Nevermind"
Multiple JOINS with WHERE clause	<pre>SELECT S.SongTitle, Als.TrackNumber, Als.TrackLength FROM Songs AS S JOIN AlbumSongs AS Als ON S.SongID =Als.SongID JOIN Albums as A ON Als.AlbumID = A.AlbumID JOIN Bands as B ON A.BandID =B.BandID WHERE B.BandName = "t.A.T.u.";</pre>	Selects the title, length and track number of all songs by "t.A.T.u"
Conceptual SELECT with GROUP BY clause	<pre>SELECT AlbumTitle FROM Albums GROUP BY AlbumYear;</pre>	Selects the title of one album created in each year in no assured order
Simple aggregate function with GROUP BY clause	<pre>SELECT AlbumYear, COUNT(*) AS NumberOfAlbums FROM Albums GROUP BY AlbumYear;</pre>	Returns a list of years and the corresponding number of albums created in each
Simple aggregate function with JOIN and GROUP BY clauses	<pre>SELECT S.SongTitle, COUNT(*) AS NumberOfVersions FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID =A.SongID GROUP BY S.SongID;</pre>	Returns a list of song titles and the number of versions of each song
Aggregate function conditions	<pre>SELECT A.AlbumTitle, AVG(Als.TrackLength) as AvgTrackLength FROM Albums AS A JOIN AlbumSongs AS Als ON A.AlbumID = Als.AlbumID GROUP BY A.AlbumID HAVING AVG(Als.TrackLength) > 360;</pre>	List the titles of all albums having an average track length longer than 6 minutes

Query Type	Syntax Example	Notes
Aggregate function conditions	<pre>SELECT A.AlbumTitle, COUNT(Als.TrackLength) as NumberOfTracks FROM Albums AS A JOIN AlbumSongs AS Als ON A.AlbumID = Als.AlbumID GROUP BY A.AlbumID HAVING COUNT(*) > 10;</pre>	List the titles of all albums containing more than 10 tracks
GROUP BY multiple columns with aggregate function condition	<pre>SELECT B.BandName, A.AlbumYear, COUNT(*) FROM Bands AS B JOIN Albums AS A ON B.BandID = A.BandID GROUP BY B.BandID, A.AlbumYear HAVING COUNT(*) > 1;</pre>	List all band names which released more than 1 album in a year
SELECT queries in SELECT queries	<pre>SELECT S.SongTitle, A.TrackLength FROM Songs AS S JOIN AlbumSongs AS A ON S.SongID = A.SongID WHERE A.TrackLength = (SELECT MAX(A.TrackLength) FROM AlbumSongs AS A);</pre>	List all song titles with the maximal track length
SELECT queries in SELECT queries with temporary tables	<pre>SELECT B.BandName FROM Bands AS B JOIN BandMusicians AS BM ON B.BandID = BM.BandID GROUP BY B.BandID HAVING COUNT(*) > (SELECT AVG(T.NumberOfMusicians) FROM (SELECT COUNT(*) AS NumberOfMusicians FROM BandMusicians AS BM GROUP BY BM.BandID) AS T);</pre>	List all band names where the band size is larger than average
SELECT queries in SELECT queries with temporary tables	<pre>SELECT A.AlbumTitle, A.AlbumYear, T.BandName FROM Albums AS A JOIN (SELECT B.BandID, B.BandName, A.AlbumYear FROM Bands AS B JOIN Albums AS A ON B.BandID = A.BandID GROUP BY B.BandID, A.AlbumYear HAVING COUNT(*) > 1) AS T ON T.BandID = A.BandID AND T.AlbumYear = A.AlbumYear;</pre>	Lists the title, year, and corresponding band name of albums released in the same year by the same band

Query Type	Syntax Example	Notes
Simple INSERT	<pre>INSERT INTO Songs VALUES (314159265, 'Canon');</pre>	Inserts 'Canon' associated with the SongID, 314159265 into Songs
Simple INSERT with specified columns	<pre>INSERT INTO Songs (SongTitle) VALUES ('Passacaglia');</pre>	Inserts 'Passacaglia' associated with the default auto-constructed SongID into Songs
SET variable and LAST_INSERT_ID	<pre>SET @songID = LAST_INSERT_ID();</pre>	Stores the last inserted primary key id into a newly declared variable songID; note: this is <i>not standard SQL</i>
Simple INSERT with a variable	<pre>INSERT INTO Albums (AlbumTitle, AlbumYear, BandID) VALUES ('Classical Music', 2012, @bandID);</pre>	Insert using an album with the title 'Classical Music' the release date 2012 and the defined BandID; note: this is <i>not standard SQL</i>
INSERT with a SELECT query	<pre>INSERT INTO AlbumSongs (SongID, AlbumID) SELECT S.SongID, A.AlbumID FROM Songs AS S JOIN Albums AS A ON S.SongTitle = 'Passacaglia' AND A.AlbumTitle = 'Classical Music';</pre>	Selects the correct AlbumID and SongID to connect in the AlbumSongs table

Query Type	Syntax Example	Notes
Simple UPDATE	<pre>UPDATE Albums SET AlbumYear = 2011 WHERE AlbumTitle = 'Classical Music';</pre>	Updates the release date of the album titled 'Classical Music' to 2011
Simple UPDATE	<pre>UPDATE Albums SET AlbumYear = AlbumYear - 1;</pre>	Reduced the release dates of <i>all</i> albums by 1 year

Query Type	Syntax Example	Notes
Simple DELETE	<pre>DELETE FROM Songs WHERE SongID = 314159265;</pre>	Deletes 'Canon' associated with the SongID, 314159265 into Songs
Simple DELETE	<pre>DELETE FROM Albums WHERE Albums.AlbumTitle = 'Classical Music' AND Albums.AlbumYear = 2012;</pre>	Deletes all albums titled 'Classical Music' released in 2012
Simple DELETE of everything in a table (careful!)	<pre>DELETE FROM Songs;</pre>	Deletes all records in the Songs table