Queries:

Query	Places can use Subqueries	Query using Join	Query Using Union	Query Using Intersection
SELECT <list of col>	FROM (Non-correlated only)	SELECT <list of col>	SELECT <list of col>	SELECT <list of col>
FROM <table>	WHERE	FROM <table1>	FROM <table1>	FROM <table1>
[WHERE <condition>]	HAVING	[INNER / OUTER] JOIN <table2>	...	...
[GROUP BY <col tuple>]	SELECT (Correlated only)	[ON <join condition>] AS <alias>	UNION [ALL]	INTERSECT [ALL]
[HAVING <condition>]		...	SELECT <list of col>	SELECT <list of col>
[ORDER BY <col tuple>]			FROM <table2>	FROM <table2>
[LIMIT <number of rows>]			...	...

Query using WITH	Query Using Correlated Subquery	Correlated Subquery in SELECT	Query using CASE
WITH <temp table> [(<temp cols>)]	SELECT ...	SELECT ... , (SELECT ...	SELECT [<list of col>,]
AS <non-correlated subquery>	FROM <table1> AS **t1**	FROM <table2> AS **t2**	CASE [t1.<col>]
SELECT ...	WHERE <predicate>	WHERE **t1**.<col1> <predicate> **t2**.<col2>),	WHEN <condition> THEN <value>
/* can use the temp table	(SELECT ...	<rest of columns>	WHEN <condition2> THEN <value2>
where ever a table can be used*/	FROM <table2> AS **t2**	FROM <table1> AS **t1**	...
	WHERE **t1**.<col1> <predicate> **t2**.<col2>	...);	[ELSE <value3>]
	);		END [AS <col name>] [, <list of cols>]
			FROM <table> AS t1
			...

Other Statements:

Insert data into a table	Delete row(s) from table	Update values in a table	Create a table
INSERT INTO <table> (<col tuple>)	DELETE FROM <table>	UPDATE <table>	CREATE TABLE <table>
VALUES (<value(s) tuple>)	[WHERE <condition>]	SET <col1>=<value1>, <col2>=<value2>, ...	(<colname> <datatype>,
	Note: If where is missing then	[WHERE <condition>]	<colname2> <datatype2>,
	delete all rows	**Note**: If WHERE missing, set all rows to values	...)

Remove table/Remove database	Grant statement privileges for a user	Revoke statement privileges for a user
DROP TABLE <table>;	GRANT <privileges> ON ’<database>’.*	REVOKE <privileges> ON ’<database>’.*
DROP DATABASE <db>;	to ’<user>’@’<host>’ identified by ’<password>’;	to ’<user>’@’<host>’
	Note: Also creates user if doesn’t exist	

Useful commands:

CREATE DATABASE <dbname>;	(as root) create database
USE <database>;	start using <database>
DESC <table>;	display meta-data for table
SHOW TABLES;	Show tables for database
RENAME TABLE/INDEX/VIEW <table/index/view name> TO <new name>;	Rename an table, index, or view to a new name
SELECT ... <col> AS <alias>	Assign <alias> for column
<table> AS <alias>	Assign <alias> for <table/view/subquery> (done in a SELECT statement)
$ mysqladmin -u <login> -p’<oldpassword>’ password <newpass>	(In terminal) Change password for <login>

Notes:

- Can specify fields from table by <table or alias>.<field>
  - This helps when there are multiple tables used or a subquery (with an alias) is used
- The **usual** order of execution of a SELECT clause: **FROM**, **ON**, **OUTER**, **WHERE**, **GROUP BY**,**ROLLUP | CUBE**, **HAVING**, **SELECT**, **DISTINCT**, **ORDER BY**, **LIMIT | TOP**

DataType		Description
TINYINT		Integer: $-2^7 : 2^7-1$ (SIGNED) or $0 : 2^8-1$ (UNSIGNED)
SMALLINT		Integer: $-2^{15} : 2^{16}-1$ (SIGNED) or $0 : 2^{16}-1$ (UNSIGNED)
MEDIUMINT		Integer: $-2^{23} : 2^{23}-1$ (SIGNED) or $0 : 2^{24}-1$ (UNSIGNED)
INT, INTEGER		Integer: $-2^{31} : 2^{31}-1$ (SIGNED) or $0 : 2^{32}-1$ (UNSIGNED)
BIGINT		Integer: $-2^{31} : 2^{31}-1$ (SIGNED) or $0 : 2^{32}-1$ (UNSIGNED)
DEC, DECIMAL, NUMERIC	(p=10,s=0)	Fixed point number: p=precision, s=decimal places
FLOAT, REAL	(m=10,d=2)	Floating point number: m=display length, d=# of decimals shown
DOUBLE, DOUBLE PRECISION	(m=16,d=4)	Floating point number: m=display length, d=# of decimals shown
DATE		Date in the format 'YYYY-MM-DD'
DATETIME		Date-time in the format 'YYYY-MM-DD HH:MM:SS'
TIMESTAMP		Date-time in the format 'YYYY-MM-DD HH:MM:SS'
YEAR		Year in the format 'YYYY'
TIME		Time in the format 'HHH:MM:SS'
CHAR	(p)	Fixed length string: p=fixed length
VARCHAR	(p)	Variable length string: p=max length
MySQL		
TINYTEXT		Text using at most 2^8-1 =255 bytes
TEXT		Text using at most $2^{16}-1$ =65,535 bytes
MEDIUMTEXT		Text using at most $2^{24}-1$ =16,777,215
LONGTEXT		Text using at most $2^{32}-1$ =4,294,967,295 bytes

Operator	Description	Example
=	Equal to	WHERE gender = 'M'
<>, !=	Not equal to	WHERE gender <> 'M'
>	Greater than	WHERE num > 5
<	Less than	WHERE num < 5
>=	Greater than or equal to	WHERE num >= 5
<=	Greater than or equal to	WHERE num <= 5
IS NULL	Value is NULL	WHERE num IS NULL
IS NOT NULL	Value is not NULL	WHERE num IS NOT NULL
BETWEEN	Between an inclusive range	WHERE num BETWEEN 3 and 5
IN	Value in a list of values	WHERE num IN (3, 5, 8)
LIKE	Search for a pattern	WHERE str LIKE 'F%'
EXISTS	Does subquery have any rows	WHERE EXISTS (<subquery>)
MySQL		
REGEXP, RLIKE	(MySQL) Search for a regular expression pattern	WHERE str RLIKE '^[FG]'

Aggregate Function	Return value
AVG(<numeric col>)	Average of non-null values
COUNT(<col or *>)	Count of non-null values
MAX(<numeric col>)	Maximum value of column
MIN(<numeric col>)	Minimium value of column
SUM(<numeric col>)	Sum of column
MySQL	
STD(<numeric col>), STDDEV_POP(<numeric col>), STDDEV(<numeric col>)	Population standard deviation
STDDEV_SAMP(<numeric col>)	Sample standard deviation
VAR_POP(<numeric col>), VARIANCE(<numeric col>)	Population variance
VAR_SAMP(<numeric col>)	Sample variance estimate
GROUP_CONCAT(<string col>)	A concatenated string

Using R with SQL (DBI package)

Function	Description	Example
dbDriver	Driver specifying the operations for creating connections to SQL Servers	m = dbDriver("MySQL"); m=MySQL()
dbConnect	Connect to a DBMS	conn = dbConnect(m, user="user", password="pass", db="database", host="hostServer")
dbSendQuery	Submits and executes SQL statement (information retrieved using fetch)	q = dbSendQuery(conn, statement="SQL_Statement")
dbGetQuery	Submits, executes SQL, and retrieves records	res = dbGetQuery(conn, statement="SQL_Statement")
fetch	Get records from a dbSendQuery	res = fetch(q, n=max.row.size)
dbCommit	Commit/rollback SQL transactions	
dbGetInfo	Get meta-data for DBIObjects	meta = dbGetInfo(q)
dbListTables	List tables in database connection	tables = dbListTables(conn)