Database Key:

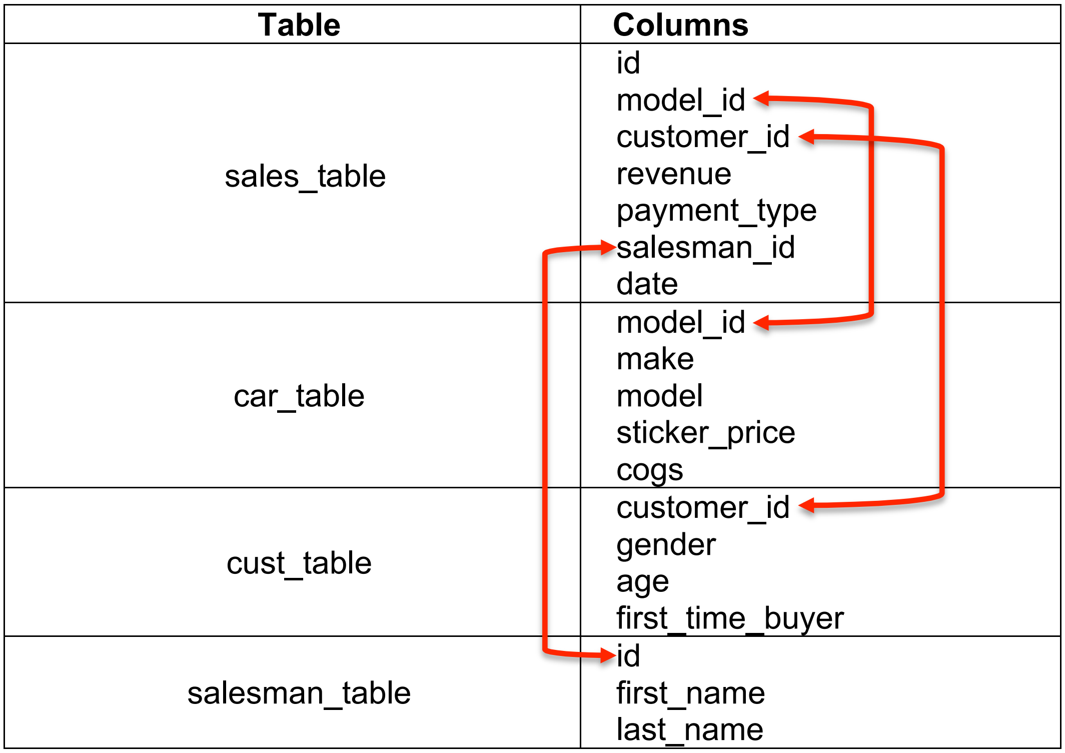

Table	Columns
sales_table	id model_id customer_id revenue payment_type salesman_id date
car_table	model_id make model sticker_price cogs
cust_table	customer_id gender age first_time_buyer
salesman_table	id first_name last_name

Strict Order of Clauses for Single Non-Union Query:

```
SELECT
FROM
     JOIN…ON
WHERE
GROUP BY
     HAVING
ORDER BY
LIMIT
```

Strict Order of Clauses for Query with Union:

```
SELECT
FROM
     JOIN…ON
WHERE
GROUP BY
     HAVING
UNION [or UNION ALL]
[second query]
ORDER BY
LIMIT
```

Example structure of a simple query:

```
SELECT
     X.column_a AS alias_a,
     SUM(Y.column_b) AS alias_b
FROM
     table_x AS X
     JOIN table_y AS Y ON X.column_a = Y.column_b
WHERE
     alias_a IN (‘some_word’,’another_word’)
     AND Y.column_b > N
GROUP BY
     alias_a HAVING alias_b > N
ORDER BY
     alias_a
LIMIT N
```

Breakdown of Clauses:

SELECT all columns FROM

```
SELECT
     *
FROM
     table_x
```

SELECT specific columns

```
SELECT
     column_a,
     column_b
FROM
     table_x
```

SELECT DISTINCT values from a column

```
SELECT
     DISTINCT column_a
FROM
     table_x
```

WHERE

```
SELECT
     column_a
FROM
     table_x
WHERE
     column_a = N
```

Some possibilities for the WHERE clause:

Expression	**Function**
= 'some_text'	Put text in quotations. Words must be capitalized according to how they appear in the database
!= x	Values DO NOT equal x
< x	Values are less than x
<= x	Values are less than or equal to x
IN (x, y)	Values are EITHER x OR y
NOT IN (x,y)	Values are NEITHER x NOR y
BETWEEN x AND y	Values are between x and y
AND	Use AND to add more expressions in WHERE clause
OR	Use OR sparingly
(column_a = x AND column_b = y) OR (column_c = z)	Use parentheses to create complex AND/OR statements

WHERE LIKE

```
     SELECT
          column_a
     FROM
          table_x
     WHERE column_a LIKE ‘%text or number%’
```

ORDER BY

```
     SELECT
          column_a
     FROM
          table_x
     ORDER BY
          column_a DESC
```

LIMIT

```
     SELECT
          column_a
     FROM
          table_x
     LIMIT N
```

JOINS

```
     SELECT
          *
     FROM
          table_x
     JOIN
          table_y ON table_x.column_a = table_y.column_a
```

Or use aliases for the tables:

```
SELECT
     *
FROM
     table_x  AS X
JOIN
     table_y AS Y ON X.column_a = Y.column_a
```

Other types of JOINS:

```
LEFT JOIN
OUTER JOIN
```

FUNCTIONS

```
SELECT
     SUM(column_a
FROM
     table_x
```

FUNCTION	**PURPOSE**
`SUM(column_a)`	Returns the sum of all values
`AVG(column_a)`	Returns average
`ROUND(AVG(column_a), 2)`	Returns average rounded to 2 decimal points
`COUNT(column_a)`	Counts the number of rows
`MAX(column_a)`	Returns the maximum value
`MIN(column_a)`	Returns the minimum value
`GROUP_CONCAT(column_a)`	Returns a comma separated list of values

GROUP BY

```
SELECT
     column_a,
     SUM(column_b)
FROM
     table_x
GROUP BY
     column_a
```

HAVING in GROUP BY Clause

```
SELECT
     column_a,
     SUM(column_b) AS alias_b
FROM
     table_x
GROUP BY
     column_a HAVING alias_b > x
```

ROLLUP

```
SELECT
     column_a,
     SUM(column_b)
FROM
     table_x
GROUP BY
     ROLLUP(column_a)
```

CASE WHEN

```
SELECT
     CASE WHEN column_a = x THEN some_value
          WHEN column_a = y THEN some_value2
          ELSE some_other_value
          END some_alias
FROM
     table_name
```

UNION

```
SELECT
     column_a
FROM
     table_x

UNION [or UNION ALL]

SELECT
     column_b
FROM
     table_y
```