SELECT * with additional columns

SELECT * FROM TABLE;

SELECT *, CALC_COLUMN FROM TABLE; ❌

- This works on MS SQL Server and MySQL but not on Oracle SQL

How to qualify column names

QUALIFIED COLUMN

UNQUALIFIED COLUMN

```
SELECT
        TABLE_NAME.COLUMN_1,
        COLUMN_2
FROM
        TABLE_NAME;
```

```
SELECT
        TABLE_NAME.*
FROM
        TABLE_NAME;
```

Qualify columns with aliased Table

```
SELECT
        COLUMN_NAMES,
FROM
        TABLE_NAME AS ❌;
```

```
SELECT
        COLUMN_NAMES,
FROM
        TABLE_NAME A;
```

Qualify columns with aliased Table

SELECT

    A.COLUMN_1,

    A.COLUMN_2,

    …

    A.COLUMN_N

FROM

    TABLE_NAME A;

SELECT

    A.*

FROM

    TABLE A;

ALIASED TABLE

Qualified vs Unqualified objects

- Unqualified object names are implicitly qualified e.g. when you select column1 from table1 it is implicitly implied that column1 is a part of table1
  - Hence an unqualified name is permitted in contexts where interpretation of the name is unambiguous.
- Qualifying table names removes ambiguity
  - A qualified name includes at least one qualifier to clarify the interpretive context by overriding a default context or providing missing context.

Qualified columns

- Where possible select only the columns you need

- Qualify columns to remove ambiguity

- Alias Table Names for code readability