

University of California Data Warehouse (UCDW)

Do's and Don'ts

Do's:

1. The best way to access UCDW as a beginner is to use the data marts. As you advance, you can venture into star schema. All data marts can be found in the IRAP_BI schema. All data mart table names contain the suffix _DM (e.g. TABLENAME_DM).
2. When using the star schema, pay attention to the fact and dimension tables. All fact table names contain the suffix _F (e.g. TABLENAME_F) and dimension table names contain the suffix _D (e.g. TABLENAME_D). A star is comprised of the fact table in the center surrounded by its corresponding dimensions.
3. A fact is connected to multiple dimensions. A SQL query will be used to join one fact to the desired dimensions around it.
4. Always use the drill across functionality to select data from multiple stars. Write separate SQL queries for each star, and join the individual SQL queries using the drill across functionality.
5. Pay attention to the grain of the fact tables when writing drill across SQL queries. For example, if you are drilling across one star to another where star 1 is at a higher grain than star 2 note that the selected column values from the star at a higher grain (i.e., star 1) will be repeated for each row from the star at the lower grain (i.e., star 2). This may not be desirable when performing counts or other functions.
6. Always use ANSI inner join and surrogate keys to join a fact to its dimension tables.

```

SELECT      fact.col_1, fact.col_2, dim.col_1, dim.col_2 ...
FROM        table_f fact
INNER JOIN  table_d dim
ON          fact.key = dim.key

```

7. When multiple fact tables are joined using the drill across functionality, it is preferred that surrogate keys are used in joins using conformed dimensions. There are some instances where the natural keys can be used such as when joining across schemas where data sources may not be conformed.
8. Use ANSI joins when joining different stars in a drill across. Commonly used ANSI joins are: Inner Join, Left/Right Outer Join, Full outer join.
9. If you need to write a big query; write the SQL code in chunks, and test each chunk along the way.
10. Test your SQL code with a small set of records first (using "fetch first 10 rows only" clause).
11. When writing a SQL query, it's a good practice to always end it with the line "FOR READ ONLY WITH UR", to avoid locking the database tables that are being accessed. The database places a read lock on every row that it is retrieving and such locks will cause other operations to wait on the lock to be released first; thus causing poor performance.

12. Study the data model and data dictionary for each subject area before writing queries so that you're familiar with the tables, their purpose and their relationships.
13. Refer to the training material and provide feedback on the topics with which you need more help.
14. Get familiar with views (suffix of _V) and MQTs (suffix of _M). They are created for your convenience and can be joined similar to the tables on which they are based.
15. When looking to do DISTINCT headcounts within the student environment using the ENROLLMENT_F fact TABLE or the SUMMER_ENROLLMENT_F FACT TABLE, always indicate in your selection criteria that the ENRL_MAJ_CMPNT_NUM and the SUMR_ENRL_MAJ_CMPNT_NUM must be equal to 1 so a student with multiple major is only counted once against the primary major.
16. To uniquely identify a student using the student dimension (STUDENT_D), always concatenate the student identifier with the campus location (STUD_LOC_CMP_CD || STUD_ID).

Don'ts:

1. Never join a fact table to another fact table directly without using the drill across functionality.
2. Never use columns from fact tables to join one fact table to another fact table.
3. Never join a data mart to a star schema or vice versa. Data marts are created for ease of use and should not be joined to any other object.
4. Do not join a data mart to data mart as it has severe data performance issues.
5. Never join unconnected dimension tables together directly except through linker or bridge tables. An example of this can be seen with the student ethnicity infrastructure.
6. Never perform aggregate functions (e.g. COUNT, SUM, and AVG) on surrogate keys. Aggregate functions must always be performed against natural keys. For example use COUNT (STUD_ID) and not COUNT (STUD_KEY). Surrogate keys should ONLY be used to join a dimension to a corresponding fact table.