

Models

CODE USED IN THE LESSON

DIM LISTINGS

models/dim/dim_listings_cleansed.sql:

```
WITH src_listings AS (  
  SELECT  
    *  
  FROM  
    {{ ref('src_listings') }}  
)  
SELECT  
  listing_id,  
  listing_name,  
  room_type,  
  CASE  
    WHEN minimum_nights = 0 THEN 1  
    ELSE minimum_nights  
  END AS minimum_nights,  
  host_id,  
  REPLACE(  
    price_str,  
    '$'  
  ) :: NUMBER(  
    10,  
    2  
  ) AS price,  
  created_at,  
  updated_at  
FROM  
  src_listings
```

DIM HOSTS

models/dim/dim_hosts_cleansed.sql:

```
{{  
  config(  
    materialized = 'view'  
  )  
}}  
  
WITH src_hosts AS (  
  SELECT  
    *  
  FROM  
    {{ ref('src_hosts') }}  
)  
SELECT  
  host_id,  
  NVL(  
    host_name,  
    'Anonymous'  
  ) AS host_name,  
  is_superhost,
```

```
    created_at,  
    updated_at  
FROM  
    src_hosts
```

EXERCISE

Create a new model in the `models/dim/` folder called `dim_hosts_cleansed.sql`.

- Use a CTE to reference the `src_hosts` model
 - SELECT every column and every record, and add a cleansing step to `host_name`:
 - If `host_name` is not null, keep the original value
 - If `host_name` is null, replace it with the value 'Anonymous'
 - Use the `NVL(column_name, default_null_value)` function
- Execute `dbt run` and verify that your model has been created

SOLUTION

```
WITH src_hosts AS (  
    SELECT  
        *  
    FROM  
        {{ ref('src_hosts') }}  
)  
SELECT  
    host_id,  
    NVL(  
        host_name,  
        'Anonymous'  
    ) AS host_name,  
    is_superhost,  
    created_at,  
    updated_at  
FROM  
    src_hosts
```

INCREMENTAL MODELS

The `fct/fct_reviews.sql` model:

```
{{  
    config(  
        materialized = 'incremental',  
        on_schema_change='fail'  
    )  
}}  
WITH src_reviews AS (  
    SELECT * FROM {{ ref('src_reviews') }}  
)  
SELECT * FROM src_reviews  
WHERE review_text is not null  
  
{% if is_incremental() %}  
    AND review_date > (select max(review_date) from {{ this }})  
{% endif %}
```

Get every review for listing 3176:

```
SELECT * FROM "AIRBNB"."DEV"."FCT_REVIEWS" WHERE listing_id=3176;
```

Add a new record to the table:

```
INSERT INTO "AIRBNB"."RAW"."RAW_REVIEWS"  
VALUES (3176, CURRENT_TIMESTAMP(), 'Zoltan', 'excellent stay!', 'positive');
```

Making a full-refresh:

```
dbt run --full-refresh
```

DIM LISTINGS WITH HOSTS

The contents of `dim/dim_listings_w_hosts.sql`:

```
WITH  
l AS (  
  SELECT  
    *  
  FROM  
    {{ ref('dim_listings_cleansed') }}  
) ,  
h AS (  
  SELECT *  
  FROM {{ ref('dim_hosts_cleansed') }}  
)  
  
SELECT  
  l.listing_id,  
  l.listing_name,  
  l.room_type,  
  l.minimum_nights,  
  l.price,  
  l.host_id,  
  h.host_name,  
  h.is_superhost as host_is_superhost,  
  l.created_at,  
  GREATEST(l.updated_at, h.updated_at) as updated_at  
FROM l  
LEFT JOIN h ON (h.host_id = l.host_id)
```

DROPPING THE VIEWS AFTER EPHEMERAL MATERIALIZATION

```
DROP VIEW AIRBNB.DEV.SRC_HOSTS;  
DROP VIEW AIRBNB.DEV.SRC_LISTINGS;  
DROP VIEW AIRBNB.DEV.SRC_REVIEWS;
```

