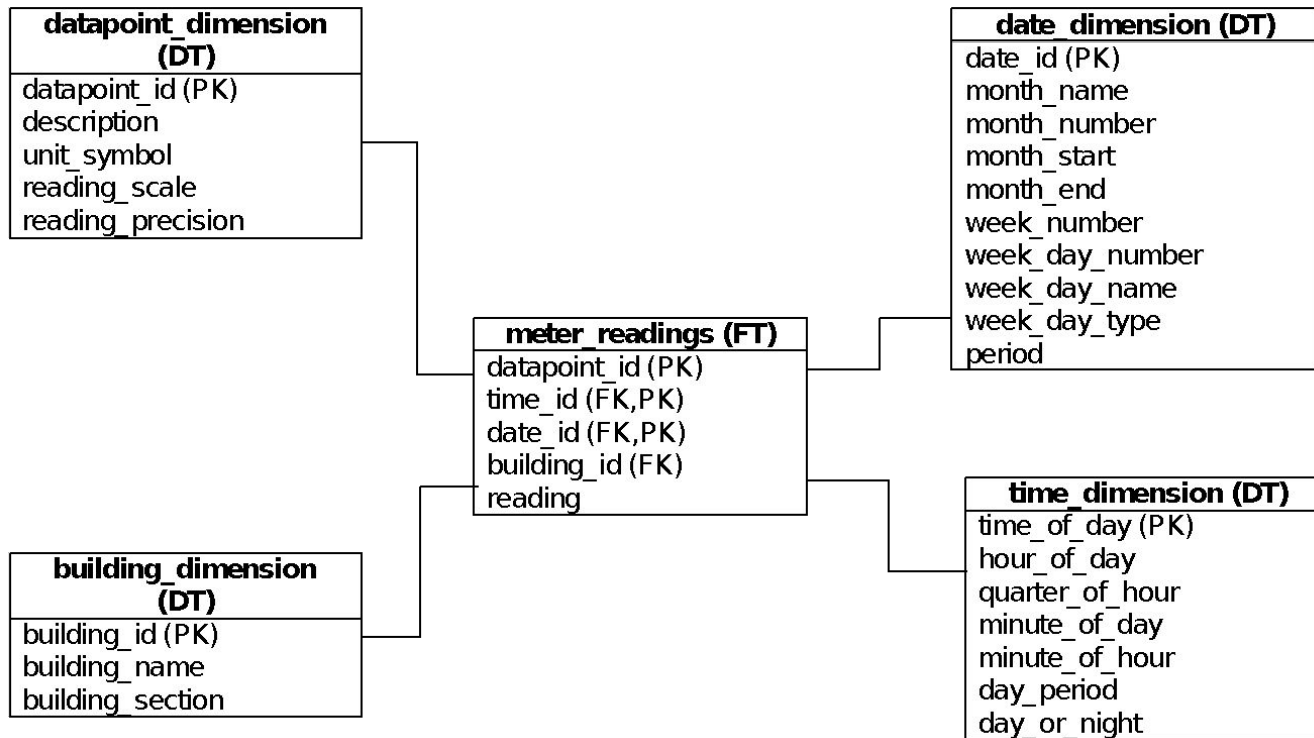


# Lab 12: Data Warehousing

Slides e Soluções do Laboratório 12

# Base de Dados Energy



# Consultas OLAP

1. Apresente uma consulta para apurar o consumo médio de todo o campus por dia da semana. Quais os dias da semana em que se regista maior e/ou menor consumo?

```
SELECT week_day_name, AVG(reading) AS avg_reading
FROM meter_readings JOIN date_dimension USING (date_id)
GROUP BY week_day_name ORDER BY avg_reading;
```

# Consultas OLAP

2. Apresente uma consulta que permita apurar o consumo médio por edifício e por semana durante as três últimas semanas do ano.

```
SELECT building_name, week_number, AVG(reading)
FROM meter_readings JOIN date_dimension USING (date_id)
JOIN building_dimension USING (building_id)
WHERE week_number >= 50
GROUP BY building_name, week_number
ORDER BY building_name, week_number;
```

# Consultas OLAP

3. Efectue o 'ROLLUP' a partir do resultado da alínea anterior, agrupando agora apenas por 'week\_number'. Pode verificar que o consumo vai caindo nas últimas semanas de Dezembro.

```
SELECT week_number, AVG(reading)
FROM meter_readings JOIN date_dimension USING (date_id)
WHERE week_number >= 50
GROUP BY week_number
ORDER BY week_number;
```

# Consultas OLAP

4. Apure que edifícios são os maiores consumidores de energia, calculando o consumo médio por edifício e ordenando o resultado.

```
SELECT building_name, AVG(reading) avg_reading
FROM meter_readings JOIN building_dimension USING (building_id)
GROUP BY building_name
ORDER BY avg_reading DESC;
```

# Consultas OLAP

5. Efectue o 'DRILL DOWN' dos resultados da alínea anterior por dia da semana (week\_day\_name) para perceber em que dias da semana é que os maiores consumidores consomem mais energia.

```
SELECT building_name, week_day_name, AVG(reading) avg_reading
FROM meter_readings JOIN date_dimension USING (date_id)
JOIN building_dimension USING (building_id)
GROUP BY building_name, week_day_name
ORDER BY avg_reading DESC;
```

# Consultas OLAP

6. Efectue o 'DRILL DOWN' dos resultados da alínea 4 por período horário (day\_period) para perceber em que períodos horários é que os edifícios maiores consumidores consomem mais energia.

```
SELECT building_name, day_period, AVG(reading) avg_reading
FROM meter_readings JOIN building_dimension USING (building_id)
JOIN time_dimension USING (time_id)
GROUP BY building_name, day_period
ORDER BY avg_reading DESC;
```



# Consultas OLAP

7. Para analisar a distribuição do consumo médio por edifício, por período horário e por hora, apresente agora os resultados do consumo médio por edifício efectuando um 'ROLLUP' sequencial sobre os campos day\_period e hour\_of\_day.

```
SELECT building_name, day_period, hour_of_day, avg(reading)
FROM meter_readings JOIN building_dimension USING (building_id)
JOIN time_dimension USING (time_id)
GROUP BY ROLLUP (building_name, day_period, hour_of_day)
ORDER BY building_name, hour_of_day;
```

ou com grouping sets

```
SELECT building_name, day_period, hour_of_day, avg(reading)
FROM meter_readings JOIN building_dimension USING (building_id)
JOIN time_dimension USING (time_id)
GROUP BY GROUPING SETS((building_name, day_period, hour_of_day),
                        (building_name, day_period), (building_name), ())
ORDER BY building_name, hour_of_day;
```

# Consultas OLAP

8. Crie uma tabela results com os resultados da alínea anterior.

```
DROP TABLE IF EXISTS results;

CREATE TABLE results AS
SELECT building_name, day_period, hour_of_day, AVG(reading) avg_reading
FROM meter_readings JOIN building_dimension USING (building_id)
JOIN time_dimension USING (time_id)
GROUP BY ROLLUP (building_name, day_period, hour_of_day)
ORDER BY building_name, hour_of_day;
```

# Consultas OLAP

9. Utilize a tabela results para determinar quais os períodos que apresentam um consumo superior à média dos consumos de todos os edifícios.

*Sugestão: Uma vez que as médias se encontram já pré-calculadas, utilize IS NULL e IS NOT NULL para obter os resultados.*

```
SELECT day_period
FROM results
WHERE hour_of_day IS NULL
      AND building_name IS NOT NULL
      AND day_period IS NOT NULL
GROUP BY day_period
HAVING AVG(avg_reading) >=
      (SELECT avg_reading
       FROM results
       WHERE building_name IS NULL);
```

# Consultas OLAP

10. Tendo por base o operador CUBE, apresente a consulta em postgres tendo como vértices building\_name, day\_period, hour\_of\_day.

```
SELECT building_name, day_period, hour_of_day, avg(reading)
FROM meter_readings JOIN building_dimension USING (building_id)
JOIN time_dimension USING (time_id)
GROUP BY CUBE (building_name, day_period, hour_of_day)
ORDER BY building_name, hour_of_day;
```