A. Which venues were the most popular?

1. By number of tickets sold

```
SELECT v.venue_id, venue_name, COUNT(*) AS tickets_sold
FROM   ticket_sales_facts
JOIN   venues_dimension AS v USING (venue_id)
GROUP BY v.venue_id, venue_name
ORDER  BY tickets_sold DESC;
```

2. By revenue (ticket sales)

```
SELECT v.venue_id, venue_name, SUM(ticket_price) AS ticket_sales_revenue
FROM   ticket_sales_facts
JOIN   venues_dimension AS v USING (venue_id)
GROUP BY v.venue_id, venue_name
ORDER  BY ticket_sales_revenue DESC;
```

B. What were the top 10 most popular bands?

3. By number of tickets sold

```
SELECT b.band_id, band_name, COUNT(*) AS tickets_sold
FROM   ticket_sales_facts
JOIN   bands_dimension AS b USING (band_id)
GROUP BY b.band_id, band_name
ORDER BY tickets_sold DESC;
```

4. By revenue (ticket sales)

```
SELECT b.band_id, band_name, SUM(ticket_price) AS ticket_sales_revenue
FROM   ticket_sales_facts
JOIN   bands_dimension AS b USING (band_id)
GROUP BY b.band_id, band_name
ORDER BY ticket_sales_revenue DESC;
```

C. What were the top 10 performances? (Since we want the band and venue name, group by the combination of band_id/band_name and venue_id/venue_name, rather than using performance_id)

5. By number of tickets sold

```
SELECT b.band_id, b.band_name, v.venue_id, v.venue_name, COUNT(*) AS tickets_sold
FROM   ticket_sales_facts
JOIN   bands_dimension AS b USING (band_id)
JOIN   venues_dimension AS v USING (venue_id)
GROUP BY b.band_id, b.band_name, v.venue_id, v.venue_name
ORDER BY tickets_sold DESC
LIMIT 10;
```

6. By revenue (ticket sales)

```
SELECT b.band_id,
       b.band_name,
       v.venue_id,
       v.venue_name,
       SUM(ticket_price) AS ticket_sales_revenue
FROM   ticket_sales_facts
JOIN   bands_dimension AS b USING (band_id)
JOIN   venues_dimension AS v USING (venue_id)
GROUP BY b.band_id, b.band_name, v.venue_id, v.venue_name
ORDER BY ticket_sales_revenue DESC
LIMIT 10;
```

D. What was the highest, lowest, and average price of a ticket? Order it by average ticket price.

7. by band (top 10)

```
SELECT b.band_id,
       band_name,
       MAX(ticket_price) AS hi,
       MIN(ticket_price) AS lo,
       AVG(ticket_price) AS average_ticket_price
FROM   ticket_sales_facts
JOIN   bands_dimension AS b USING (band_id)
GROUP BY b.band_id, band_name
ORDER BY average_ticket_price DESC
LIMIT 10;
```

8. By venue

```
SELECT v.venue_id,
       venue_name,
       MAX(ticket_price) AS hi,
       MIN(ticket_price) AS lo,
       AVG(ticket_price) AS average_ticket_price
FROM   ticket_sales_facts
JOIN   venues_dimension AS v USING (venue_id)
GROUP BY v.venue_id, venue_name
ORDER BY average_ticket_price DESC
LIMIT 10;
```

9. by performance (top 10)

```
SELECT b.band_id,
       band_name,
       v.venue_id,
       venue_name,
       MAX(ticket_price) AS hi,
       MIN(ticket_price) AS lo,
       AVG(ticket_price) AS average_ticket_price
FROM   ticket_sales_facts
JOIN   venues_dimension AS v USING (venue_id)
JOIN   bands_dimension AS b USING (band_id)
GROUP BY v.venue_id, venue_name, b.band_id, band_name
ORDER BY average_ticket_price DESC
LIMIT 10;
```

E. The AMD venue is where the music festival showcases up-and-coming bands. We want to identify those bands that sell out this venue because we may want to sign them up to play in a bigger venue in next year's festival.

Write a query that shows the performances that sold out this venue in the last five years, ordering them from most to least tickets sold. Note that the venue allows Standing Room Only (SRO) tickets to be sold, so the number of tickets sold could exceed the venue's capacity (which is the number of seats in the venue).

10. Bands that sold out the AMD venue since 2018.

```
SELECT venue_name,
       band_name,
       EXTRACT(YEAR FROM performance_start) AS year,
       capacity,
       COUNT(*) as tickets_sold
FROM   ticket_sales_facts
JOIN   performances_dimension USING (performance_id)
JOIN   bands_dimension USING (band_id)
JOIN   venues_dimension USING (venue_id)
WHERE  EXTRACT(YEAR FROM performance_start) > 2017
  AND  venue_name = 'AMD'
GROUP  BY venue_name, band_name, capacity, performance_id, year
HAVING COUNT(*) >= capacity
ORDER BY year desc, band_name;
```