```sql
select * from projectdata;

-- total episodes

SELECT MAX(ep) FROM projectdata;

SELECT COUNT(DISTINCT ep) FROM projectdata;


-- pitches

SELECT COUNT(DISTINCT brand) FROM projectdata;


--pitches converted

SELECT AVG(a.EQUITY_TAKEN_)
FROM
  (SELECT *
   FROM projectdata
   WHERE EQUITY_TAKEN_ > 0) a;

-- total male

SELECT SUM(male) FROM projectdata;

-- total female

SELECT SUM(female) FROM projectdata;

--gender ratio

SELECT SUM(female) / SUM(male) AS gender_ratio
FROM projectdata;
```

```sql
-- total invested amount

SELECT SUM(AMOUNT_INVESTED_LAKHS) AS total_invested_amount
FROM projectdata;

-- avg equity taken

SELECT AVG(a.EQUITY_ASKED_)
FROM
  (SELECT *
   FROM projectdata
   WHERE EQUITY_ASKED_ > 0) a;

--highest deal taken

SELECT MAX(AMOUNT_INVESTED_LAKHS) FROM projectdata;

--higheest equity taken

SELECT MAX(EQUITY_TAKEN_)
FROM projectdata;

-- startups having at least women

SELECT SUM(a.female_count) AS startups_having_at_least_women
FROM (
    SELECT CASE WHEN female > 0 THEN 1 ELSE 0 END AS female_count
    FROM projectdata -- Replace 'projectdata' with the correct tab
) a;

-- pitches converted having atleast ne women

SELECT *
FROM projectdata;
```

```sql
SELECT SUM(b.female_count)
FROM
  (SELECT CASE WHEN a.female > 0 THEN 1 ELSE 0 END AS female_count,
   FROM
     (SELECT *
      FROM projectdata
      WHERE deal != 'No Deal') a) b;

-- avg team members

SELECT AVG(team_members) FROM projectdata;

-- amount invested per deal

SELECT AVG(a.AMOUNT_INVESTED_LAKHS) AS amount_invested_per_deal
FROM (SELECT *
      FROM projectdata
      WHERE deal != 'No Deal') a;


-- avg age group of contestants

SELECT AVG_AGE, COUNT(AVG_AGE) AS cnt
FROM projectdata
GROUP BY AVG_AGE
ORDER BY cnt DESC;

-- location group of contestants

SELECT location, COUNT(location) AS cnt
FROM projectdata
GROUP BY location
ORDER BY cnt DESC;
```

```sql
SELECT sector, COUNT(sector) AS cnt
FROM projectdata
GROUP BY sector
ORDER BY cnt DESC;

--partner deals

SELECT partners, COUNT(partners) AS cnt
FROM projectdata
WHERE partners != '-'
GROUP BY partners
ORDER BY cnt DESC;

-- making the matrix

SELECT *
FROM projectdata;

SELECT 'Ashnner' AS keyy, COUNT(ASHNEER_AMOUNT_INVESTED)
FROM projectdata
WHERE ASHNEER_AMOUNT_INVESTED IS NOT NULL;

SELECT 'Ashnner' AS keyy, COUNT(CAST(ASHNEER_AMOUNT_INVESTED AS NUM
FROM projectdata
WHERE ASHNEER_AMOUNT_INVESTED IS NOT NULL AND ASHNEER_AMOUNT_INVES


SELECT 'Ashneer' AS keyy, SUM(C.ASHNEER_AMOUNT_INVESTED), AVG(C.ASH
FROM (SELECT *
      FROM projectdata
      WHERE ASHNEER_EQUITY_TAKEN_ != 0 AND ASHNEER_EQUITY_TAKEN_ IS
```

```
FROM (SELECT a.keyy, a.total_deals_present, b.total_deals
      FROM (SELECT 'Ashneer' AS keyy, COUNT(ASHNEER_AMOUNT_INVESTE
            FROM projectdata
            WHERE ASHNEER_AMOUNT_INVESTED IS NOT NULL) a
      INNER JOIN (SELECT 'Ashneer' AS keyy, COUNT(ASHNEER_AMOUNT_I
                  FROM projectdata
                  WHERE ASHNEER_AMOUNT_INVESTED IS NOT NULL AND ASH
          ON a.keyy = b.keyy) m
INNER JOIN (SELECT 'Ashneer' AS keyy, SUM(C.ASHNEER_AMOUNT_INVESTE
            FROM (SELECT *
                  FROM projectdata
                  WHERE ASHNEER_EQUITY_TAKEN_ != 0 AND ASHNEER_EQU
    ON m.keyy = n.keyy;


-- which is the startup in which the highest amount has been inve

SELECT brand, sector, AMOUNT_INVESTED_LAKHS
FROM (
  SELECT brand, sector, AMOUNT_INVESTED_LAKHS, RANK() OVER (PARTITI
  FROM projectdata
) c
WHERE c.rnk = 1;
```