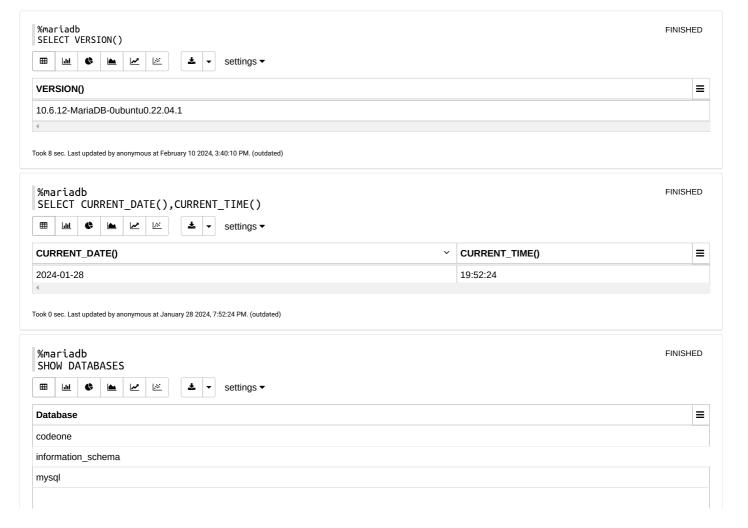
Sakila Database SQL Analysis

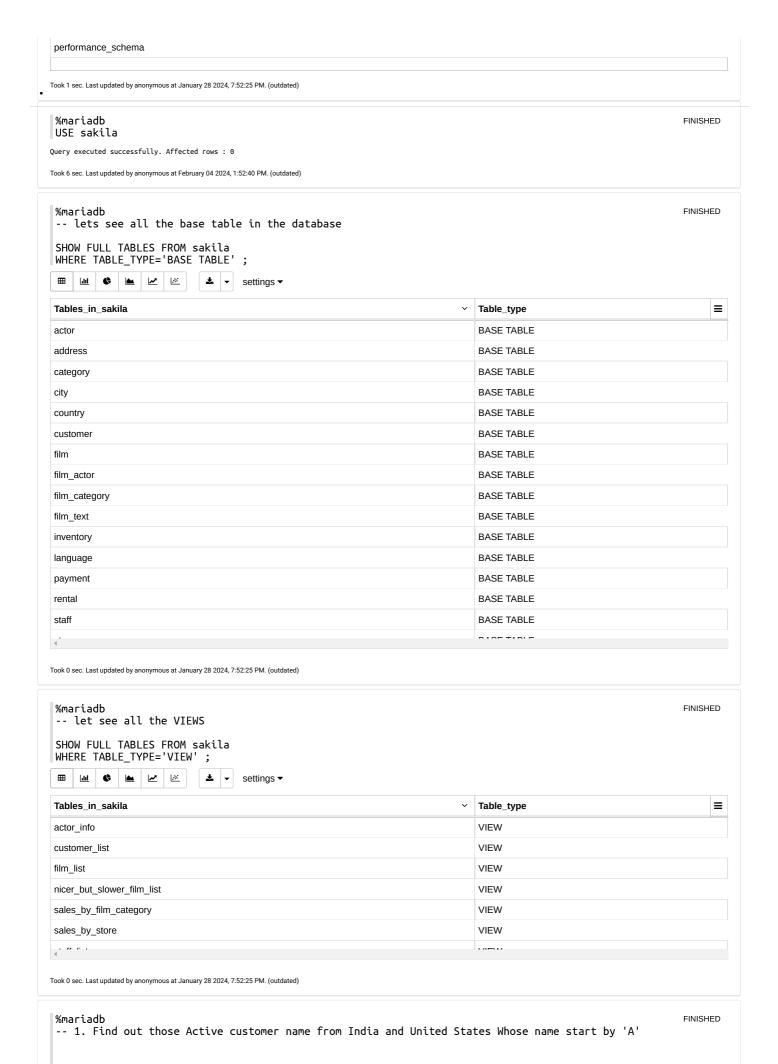
In this notebook, I perform a data analysis using the Mariadb SQL language on the Sakila database. The Sakila database is a sample database that represents a DVD rental store. The analysis aims to answer various questions related to the Sakila database and gain insights from the data.

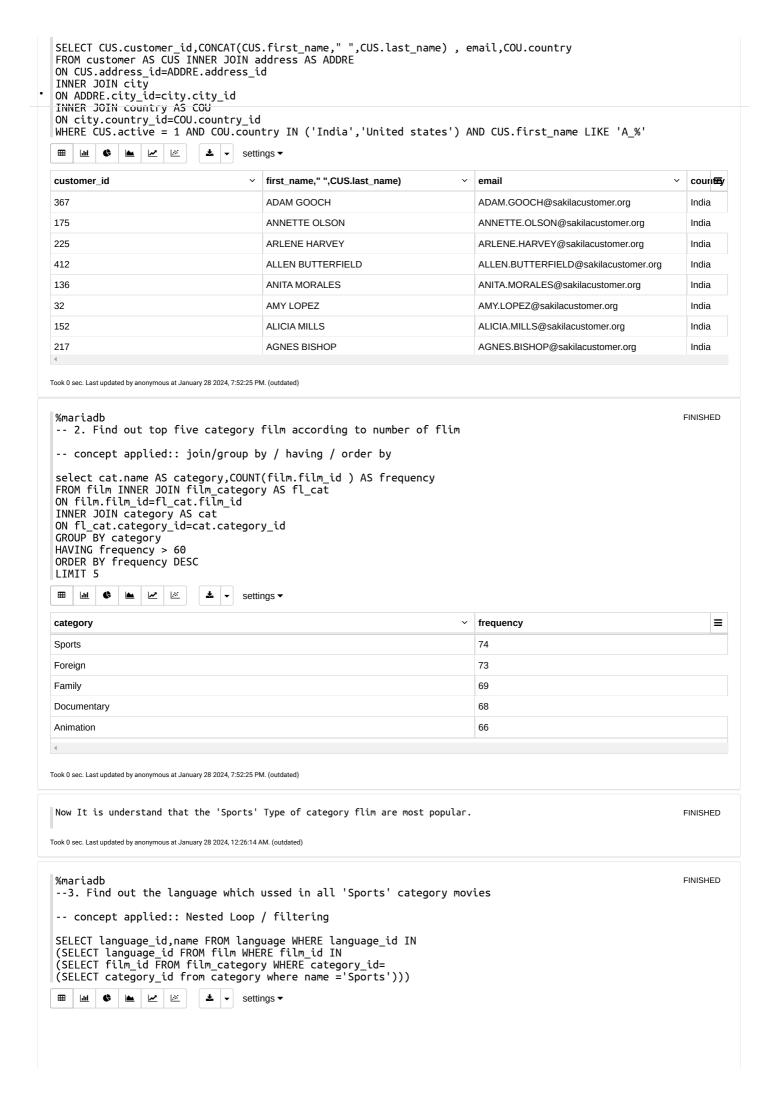
Analysis Questions

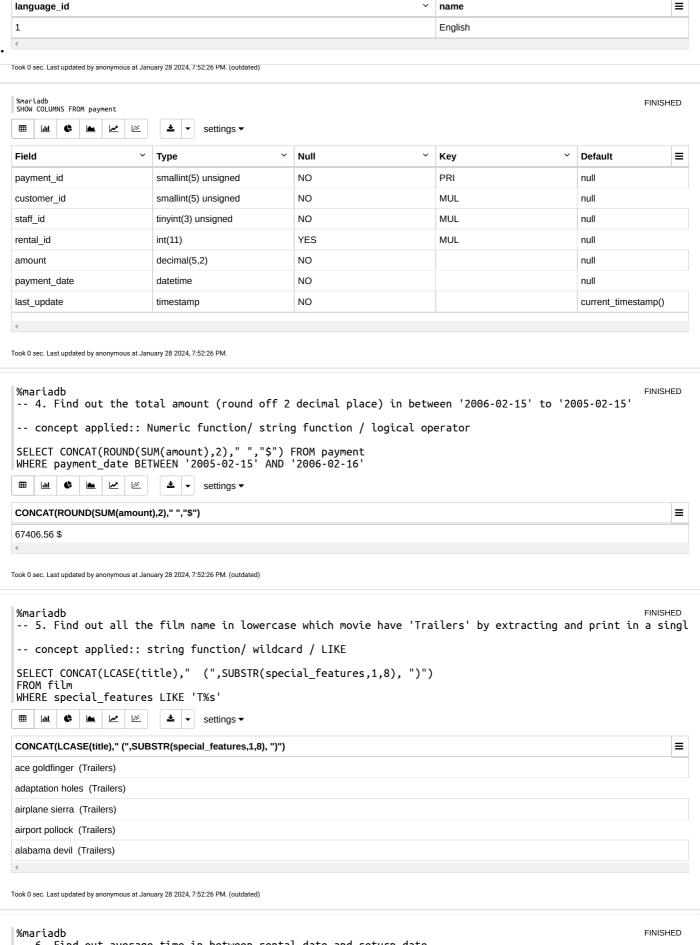
- 1. Find out those Active customer name from India and United States Whose name start by 'A'.
- 2. Find out top five category film according to number of flim.
- 3. Find out the language which ussed in all 'Sports' category movies
- 4. Find out the total amount (round off 2 decimal place) in between '2006-02-15' to '2005-02-15'
- 5. Find out all the film name in lowercase which movie have 'Trailers' by extracting and print in a single line.
- 6. Find out average time in between rental date and return date
- 7. Find out total number of rental order which yet not returned
- 8. Findout the payment_id of the rental_id =1009. Find out the film name which earn most with the amount value
- 9. The name of film and amount which earned most
- 10. Find out actor's who participate in the most earned movie 'telegraph voyage'?
- 11. Find out how many copies of film 'telegraph voyage' exist in the inventory system
- 12. Find out what is the total amount paid by each customer for all their rental?
- 13. Find out how many films from each category each store has?
- 14. Calculate the total revenue of each store
- 15. Find out top 5 customers who have rented the most number of films, and rank them based on the total number of films rented?

Took 9 sec. Last updated by anonymous at February 13 2024, 3:30:33 PM





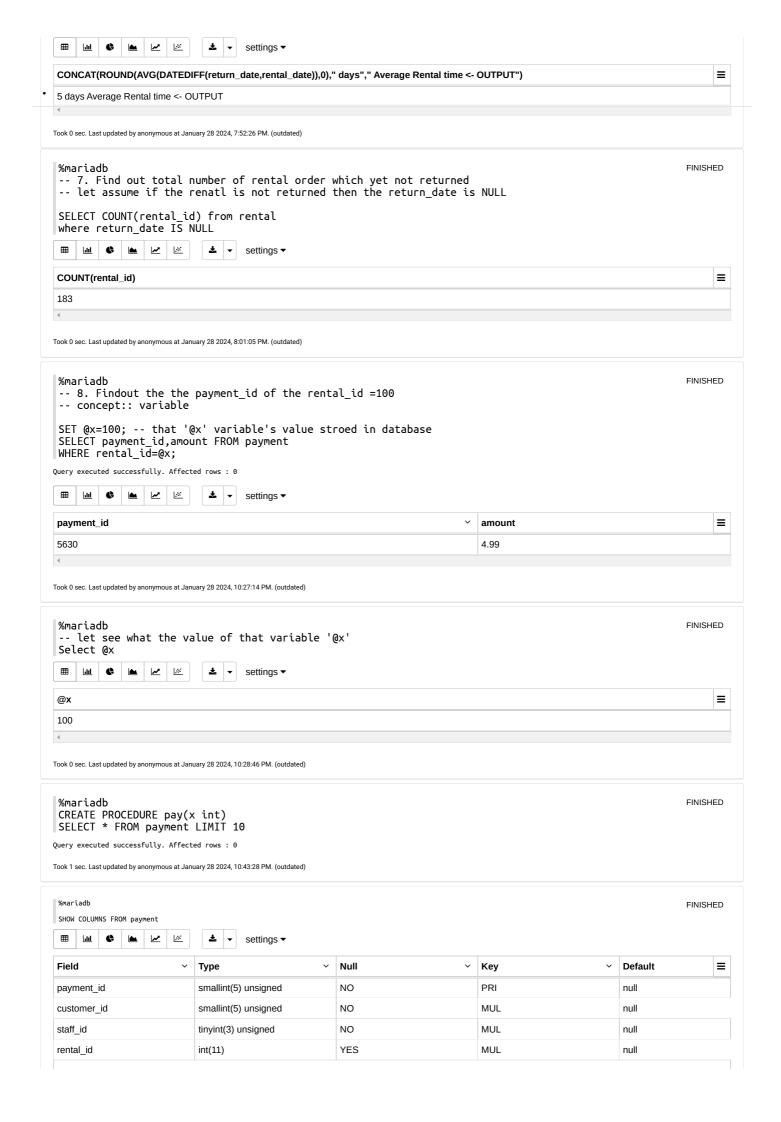




-- 6. Find out average time in between rental date and return date

-- concept applied:: date function / numeric function / string function

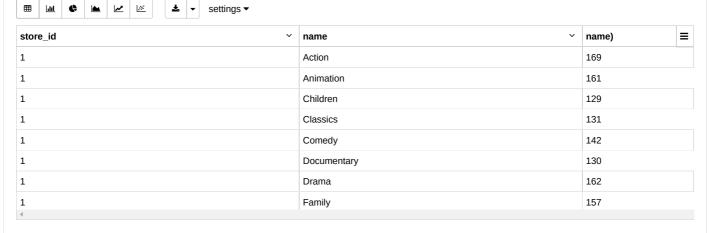
SELECT CONCAT(ROUND(AVG(DATEDIFF(return_date,rental_date)),0)," days"," Average Rental time <- OUTPUT") FROM





7 Took 0 sec. Last updated by anonymous at January 30 2024, 5:12:56 PM. (outdated) %mariadb FINISHED -- 13. Find out what is the total amount paid by each customer for all their rental ? SELECT c.customer_id,CONCAT(c.first_name," ",c.last_name),SUM(p.amount) FROM customer c INNER JOIN rental r on c.customer_id=r.customer_id INNER JOIN payment p on r.rental_id=p.rental_id
GROUP BY c.customer_id
ORDER BY SUM(p.amount) DESC latel settings ▼ customer_id first_name," ",c.last_name) amount) ≡ 221.55 526 KARL SEAL **ELEANOR HUNT** 148 216.54 CLARA SHAW 195.58 144 137 RHONDA KENNEDY 194.61 MARION SNYDER 178 194.61 459 TOMMY COLLAZO 186.62 469 WESLEY BULL 177.60 468 TIM CARY 175.61 Took 0 sec. Last updated by anonymous at January 30 2024, 5:31:50 PM. (outdated) %mariadb FINISHED -- 14. Find out how many films from each category each store has ? select s.store_id,c.name,COUNT(c.name) from store s INNER JOIN inventory i ON s.store_id=i.store_id INNER JOIN film f ON f.film_id=i.film_id INNER JOIN film_category fc ON f.film_id=fc.film_id INNER JOIN category c ON fc.category_id=c.category_id GROUP BY s.store_id,c.name

FINISHED Took 0 sec. Last updated by anonymous at January 30 2024, 5:02:40 PM. (outdated) settings



Took 0 sec. Last updated by anonymous at January 30 2024, 5:55:48 PM. (outdated)

%mariadb use sakila FINISHED

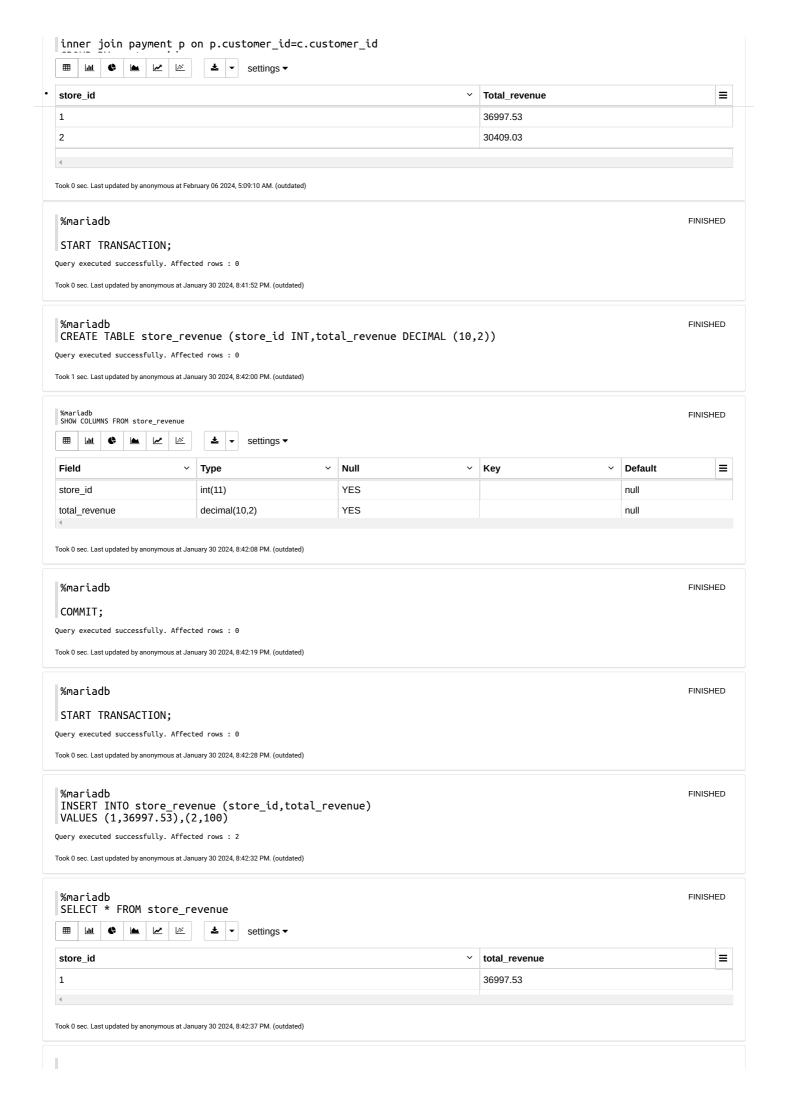
Query executed successfully. Affected rows : θ

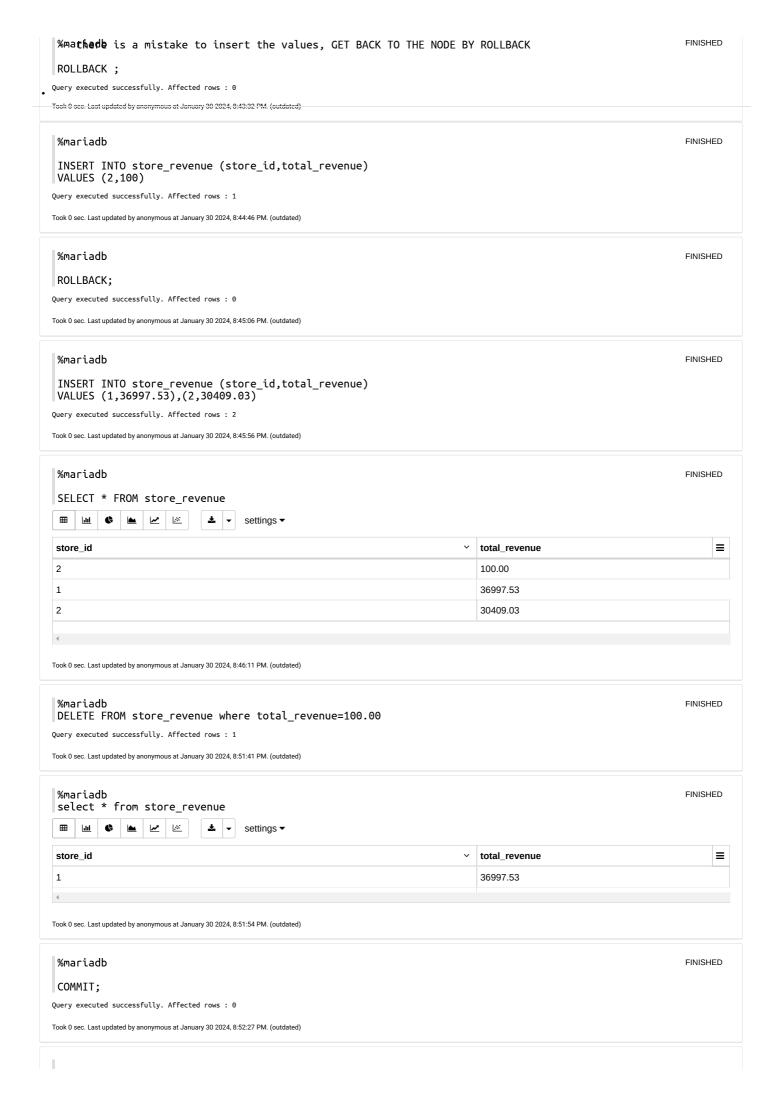
Took 0 sec. Last updated by anonymous at February 06 2024, 5:09:06 AM. (outdated)

FINISHED

-- 15. Calculate the total revenue of each store |

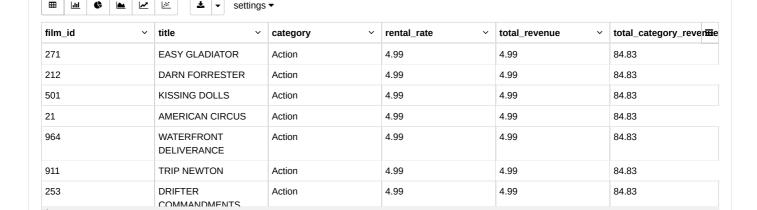
select s.store_id,SUM(p.amount) Total_revenue from store s inner join customer c on s.store_id=c.store_id





‱afiadbind out top 5 customers who have rented the most number of films, and rank them based on the 덴어턴티다 SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name), RANK() OVER (ORDER BY COUNT(r.rental_id) DESC) AS rental_rank,COUNT(r.rental_id) AS total_rentals FROM customer c JOIN rental r ON c.customer_id = r.customer_id GROUP BY c.customer_id ORDER BY rental_rank LIMIT 5; hil Ŧ settings first_name, ' ', c.last_name) rental_rank ∨ total_**±e**r customer id 148 **ELEANOR HUNT** 46 1 526 KARI SEAL 2 45 **CLARA SHAW** 3 42 3 236 MARCIA DEAN 42 TAMMY SANDERS 5 41 Took 0 sec. Last updated by anonymous at February 04 2024, 1:59:06 PM. (outdated) FINISHED %mariadb -- 17. Find out the total revenue generated by each category of films, along with the rank of each film and its category based on revenue? **SELECT** f.film_id, f.title c.name AS category, f.rental rate, SUM(f.rental_rate) OVER (partition by f.title order by f.title) AS total_revenue, SUM(f.rental_rate) OVER (PARTITION BY c.name ORDER BY f.rental_rate DESC) AS total_category_revenue, RANK() OVER (PARTITION BY c.name ORDER BY f.rental_rate DESC) AS rank_within_category

f.title,
f.title,
c.name AS category,
f.rental_rate,
SUM(f.rental_rate) OVER (partition by f.title order by f.title) AS total_reven
SUM(f.rental_rate) OVER (PARTITION BY c.name ORDER BY f.rental_rate DESC) AS t
RANK() OVER (PARTITION BY c.name ORDER BY f.rental_rate DESC) AS rank_within_c
FROM
film f
JOIN
film_category fc ON f.film_id = fc.film_id
JOIN
category c ON fc.category_id = c.category_id
ORDER BY
c.name, rank_within_category;



Took 0 sec. Last updated by anonymous at February 04 2024, 2:14:07 PM. (outdated)

%mariadb
-- 18. Find out the number of rentals, total revenue, and average rental duration for each month with each y

SELECT
 YEAR(r.rental_date) AS rental_year,
 MONTH(r.rental_date) AS rental_month,
 COUNT(r.rental_id) OVER (PARTITION BY YEAR(r.rental_date), MONTH(r.rental_date)) AS total_rentals,
 SUM(p.amount) OVER (PARTITION BY YEAR(r.rental_date), MONTH(r.rental_date)) AS total_revenue,

AVG(r.return_date - r.rental_date) OVER (PARTITION BY YEAR(r.rental_date), MONTH(r.rental_date)) AS aver FROM _ rental r JOIN payment p ON r.rental_id = p.rental_id
ORDER BY rental_year, rental_month; <u>ш</u> **♦ <u>►</u> <u>►</u>** settings rental_year rental_month v total_rentals total_revenue ≖ 2005 5 1156 4823.44 2005 5 1156 4823.44 2005 5 1156 4823.44 2005 5 4823.44 1156 5 2005 1156 4823.44 5 2005 1156 4823.44 5 2005 1156 4823.44 2005 5 1156 4823.44 Took 0 sec. Last updated by anonymous at February 04 2024, 2:27:32 PM. (outdated) %mariadb READY