

SQL-এ `ALTER` কমান্ড ব্যবহার করা হয় একটি টেবিলের কাঠামো পরিবর্তন করতে। যেমন টেবিলের নতুন কলাম যোগ করা, বিদ্যমান কলাম মুছে ফেলা, বা কলামের ধরন পরিবর্তন করা।

এসব কমান্ড আপনার টেবিলের কাঠামো পরিবর্তন করতে সাহায্য করবে।

1. নতুন কলাম যোগ করা:

```
ALTER TABLE table_name ADD column_name column_type;
```

```
ALTER TABLE employees ADD phone_number VARCHAR(15);
```

2. কলাম মুছে ফেলা:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

```
ALTER TABLE employees DROP COLUMN phone_number;
```

3. কলামের নাম পরিবর্তন করা:

```
ALTER TABLE table_name
```

```
RENAME COLUMN old_column_name TO new_column_name;
```

```
ALTER TABLE employees
```

```
RENAME COLUMN phone_number TO contact_number;
```

4. কলামের ধরন পরিবর্তন করা:

```
ALTER TABLE table_name
```

```
MODIFY COLUMN column_name new_data_type;
```

```
ALTER TABLE employees
```

```
MODIFY COLUMN salary DECIMAL(10, 2);
```

RENAME : SQL-এ একটি টেবিলের নাম পরিবর্তন করতে `RENAME` কমান্ড ব্যবহার করা হয়।

উদাহরণ:

```
RENAME TABLE old_table_name TO new_table_name;
```

```
RENAME TABLE employees TO staff_members;
```

এটি `employees` টেবিলের নাম পরিবর্তন করে `staff_members` রাখবে।

1. Select * from 'customer' where state ='nh'
2. Select state , cus_id from 'customer' where state ='nh'
3. Select state , cus_id as id from 'customer' where state ='nh'
4. Select * from 'customer' where state ='nh' AND city = ' salem'
5. Select * from 'customer' where state ='nh' AND city = ' salem' AND cus_id=5
6. Select * from 'customer' where state ='nh' AND city = ' salem' AND cus_id=5 OR cus_id=1

Same as

SELECT * FROM 'customer' WHERE (state = 'nh' AND city = 'salem' AND cus_id = 5) OR cus_id = 1;

৩. DISTINCT এবং অন্যান্য ফাংশন একত্রে ব্যবহার:

SELECT DISTINCT OPEN_BRANCH_ID as Total_BRANCH

FROM account;

Output : 1,2,3,4

এটি account টেবিল থেকে কতটি unique OPEN_BRANCH_ID আছে তা গণনা করবে।

DISTINCT ব্যবহার করে আপনি একই ধরনের মানের পুনরাবৃত্তি এড়াতে পারেন, ফলে টেবিল থেকে অনন্য ডেটা নির্বাচন করতে পারবেন।

SELECT COUNT(DISTINCT OPEN_BRANCH_ID) as Total_BRANCH

FROM account;

Output : 4 (total 4 ti branch)

এটি account টেবিল থেকে unique OPEN_BRANCH_ID গুলোর সংখ্যা গণনা করবে। DISTINCT ব্যবহার করা হয়েছে যাতে প্রতিটি OPEN_BRANCH_ID একবারই গণনা করা হয়, এমনকি যদি একই মান একাধিকবার টেবিলে উপস্থিত থাকে।

Having

আমরা সাধারনত শর্ত দিয়ে ডাটা আনতে চাইলে where ব্যবহার করি , কিন্তু

অ্যাগ্রিগেট ফাংশনের ফলাফলের উপর শর্ত দিতে হলে আপনাকে HAVING ক্লজ ব্যবহার করতে হবে।

Possible

```
SELECT *  
FROM ACCOUNT  
WHERE AVAIL_BALANCE > "100";
```

Not possible

```
SELECT *  
FROM ACCOUNT  
WHERE AVAIL_BALANCE > AVG(AVAIL_BALANCE);
```

Possible (যাদের AVAIL_BALANCE গড় BALANCE এর বেশি)

```
SELECT AVAIL_BALANCE  
FROM ACCOUNT  
GROUP BY AVAIL_BALANCE  
HAVING AVAIL_BALANCE > AVG(AVAIL_BALANCE);
```

আপনি যদি `AVG (AVAIL_BALANCE)` এর একটি নির্দিষ্ট মানের চেয়ে বেশি হলে সেই রেকর্ডগুলো দেখতে চান, তাহলে কোডটি এমন হওয়া উচিত:

```
SELECT AVAIL_BALANCE  
FROM ACCOUNT  
GROUP BY AVAIL_BALANCE  
HAVING AVG(AVAIL_BALANCE) >1000;
```

ekhane ki where clause use korte parbe ?

না, আপনি এই পরিস্থিতিতে `WHERE` ক্লজ ব্যবহার করতে পারবেন না, কারণ `WHERE` ক্লজ অ্যাগ্রিগেট ফাংশনের (যেমন `AVG()`) উপর সরাসরি শর্ত দিতে পারে না। `WHERE` ক্লজ শুধুমাত্র সারি-স্তরের শর্ত প্রয়োগ করতে ব্যবহৃত হয় এবং এটি অ্যাগ্রিগেট ফাংশনের ফলাফল যাচাই করতে পারে না।

অ্যাগ্রিগেট ফাংশনের ফলাফলের উপর শর্ত দিতে হলে আপনাকে `HAVING` ক্লজ ব্যবহার করতে হবে।

Sub query kore abar possible

```
SELECT *  
FROM ACCOUNT  
WHERE AVAIL_BALANCE > (SELECT AVG(AVAIL_BALANCE) FROM ACCOUNT);
```

`WHERE`, `GROUP BY`, এবং `HAVING` একসাথে ব্যবহার করে একটি SQL ক্যুয়ারির উদাহরণ নিচে দেয়া হলো:

- প্রতিটি শাখার জন্য গড় `AVAIL_BALANCE` নির্বাচন করবে।
- `WHERE STATUS = 'ACTIVE':` `STATUS` কলামের মান 'ACTIVE' থাকা রেকর্ডগুলো ফিল্টার করবে।
- `GROUP BY OPEN_BRANCH_ID:` প্রতিটি শাখা আইডি অনুসারে গ্রুপ তৈরি করবে।
- `HAVING AVG(AVAIL_BALANCE) > 5000:` যেসব শাখার গড় `AVAIL_BALANCE` ৫০০০ এর বেশি, সেগুলো দেখাবে।

```
SELECT OPEN_BRANCH_ID, AVG(AVAIL_BALANCE) AS avg_balance  
FROM ACCOUNT
```

WHERE STATUS = 'ACTIVE'

GROUP BY OPEN_BRANCH_ID

HAVING AVG(AVAIL_BALANCE) > 5000;

Find the N Largest Salary

SELECT MAX(`AVAIL_BALANCE`) AS max_balance

FROM account;

SELECT *

FROM account

ORDER BY `AVAIL_BALANCE` DESC

LIMIT 1;

Find 2nd highest salary

SELECT *

FROM account

ORDER BY `AVAIL_BALANCE` DESC

LIMIT 1

OFFSET 1

SELECT * FROM account

WHERE AVAIL_BALANCE = (SELECT DISTINCT(AVAIL_BALANCE)

FROM account ORDER BY account DESC LIMIT 1 OFFSET N-1);

এটি তৃতীয় সর্বোচ্চ AVAIL_BALANCE নির্বাচন করবে

```
SELECT * FROM account WHERE AVAIL_BALANCE = ( SELECT AVAIL_BALANCE FROM account ORDER BY  
AVAIL_BALANCE DESC LIMIT 1 OFFSET 2 );
```