

CONTACTS

```
SELECT cl.name AS client_name, COUNT(c.id) AS total_contacts
FROM contacts c
JOIN client cl ON c.client_id = cl.id
GROUP BY cl.name
ORDER BY total_contacts DESC;
```

```
SELECT stage, COUNT(*) AS contact_count,
ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM contacts), 2) AS stage_percentage
FROM contacts
GROUP BY stage
ORDER BY contact_count DESC;
```

```
SELECT owner, COUNT(*) AS total_contacts,
SUM(CASE WHEN stage = 'Active' THEN 1 ELSE 0 END) AS active_contacts
FROM contacts
GROUP BY owner
ORDER BY total_contacts DESC;
```

```
SELECT c.first_name, c.last_name, c.email, c.stage, cl.name AS client_name
FROM contacts c
JOIN client cl ON c.client_id = cl.id
WHERE c.stage IN ('Prospect', 'Negotiation');
```

```
SELECT cl.name AS client_name, COUNT(*) AS new_contacts_last_month
FROM contacts c
JOIN client cl ON c.client_id = cl.id
WHERE c.created_at >= DATEADD(DAY, -30, CAST(GETDATE() AS date))
GROUP BY cl.name
ORDER BY new_contacts_last_month DESC;
```

```
SELECT c.owner, COUNT(DISTINCT c.client_id) AS clients_managed, COUNT(*) AS
total_contacts
FROM contacts c
GROUP BY c.owner
ORDER BY clients_managed DESC;
```

```
SELECT c.stage, cl.industry, COUNT(c.id) AS total_contacts
FROM contacts c
JOIN client cl ON c.client_id = cl.id
GROUP BY cl.industry, c.stage
ORDER BY total_contacts DESC;
```

```
SELECT c.client_name, MAX(c.updated_at) AS last_updated_contact
FROM contacts c
GROUP BY c.client_name
ORDER BY last_updated_contact DESC;
```

```
SELECT c.first_name, c.last_name, c.email, c.stage
FROM contacts c
WHERE c.email LIKE '%@gmail.com%'
ORDER BY c.created_at DESC;
```

```
SELECT cl.name AS client_name, COUNT(c.id) AS contact_count
FROM contacts c
JOIN client cl ON c.client_id = cl.id
GROUP BY cl.name
HAVING COUNT(c.id) > 5
ORDER BY contact_count DESC;
```

```
SELECT c.stage, COUNT(DISTINCT c.client_id) AS unique_clients_in_stage
FROM contacts c
GROUP BY c.stage
ORDER BY unique_clients_in_stage DESC;
```

```
SELECT owner, MAX(updated_at) AS last_contact_update, MIN(created_at) AS
first_contact_created
FROM contacts
GROUP BY owner
ORDER BY last_contact_update DESC;
```

```
SELECT c.client_name, COUNT(DISTINCT c.owner) AS owners_involved
FROM contacts c
GROUP BY c.client_name
HAVING COUNT(DISTINCT c.owner) > 1
ORDER BY owners_involved DESC;
```

```
SELECT c.client_name, COUNT(*) AS total_contacts,
SUM(CASE WHEN c.stage = 'Closed' THEN 1 ELSE 0 END) AS closed_contacts
FROM contacts c
GROUP BY c.client_name
ORDER BY total_contacts DESC;
```

CLIENT

```
SELECT c.id AS client_id, c.name AS client_name, c.status, c.industry, c.owner_name,
MAX(p.id) AS sample_project_id
FROM client c
```

```
LEFT JOIN project p ON c.id = p.client_id
WHERE c.status = 'active'
GROUP BY c.id, c.name, c.status, c.industry, c.owner_name
ORDER BY c.owner_name;

SELECT c.id, c.name, c.email, c.status, c.industry
FROM client c
LEFT JOIN project p ON c.id = p.client_id
WHERE p.client_id IS NULL AND c.status IN ('prospect', 'inactive');

SELECT industry, COUNT(*) AS total_clients, ROUND(AVG(no_employees), 2) AS
avg_employees
FROM client
WHERE industry IS NOT NULL
GROUP BY industry
ORDER BY total_clients DESC;

SELECT FORMAT(created_at, 'yyyy-MM') AS month, COUNT(*) AS clients_created
FROM client
WHERE created_at >= DATEADD(MONTH, -12, CAST(GETDATE() AS date))
GROUP BY FORMAT(created_at, 'yyyy-MM')
ORDER BY month ASC;

SELECT timezone, COUNT(*) AS client_count
FROM client
WHERE status = 'active'
GROUP BY timezone
ORDER BY client_count DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;

SELECT id, name, email, phone
FROM client
WHERE (email IS NULL OR email = "") OR (phone IS NULL OR phone = "");

SELECT id, name, created_at, updated_at
FROM client
WHERE updated_at < created_at
ORDER BY created_at DESC;

SELECT owner_name, status, COUNT(*) AS total_clients
FROM client
GROUP BY owner_name, status
ORDER BY total_clients DESC;
```

```
SELECT owner_name, COUNT(*) AS new_clients
FROM client
WHERE created_at >= DATEADD(MONTH, -3, DATEFROMPARTS(YEAR(GETDATE()),
MONTH(GETDATE()), 1))
GROUP BY owner_name
ORDER BY new_clients DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
```

```
SELECT country, state, COUNT(*) AS active_clients
FROM client
WHERE status = 'active'
GROUP BY country, state
ORDER BY active_clients DESC;
```

```
SELECT country, industry, COUNT(*) AS client_count
FROM client
WHERE status = 'active'
GROUP BY country, industry
ORDER BY country, client_count DESC;
```

```
SELECT id, name, hubspot_id, drive_id, delta_token
FROM client
WHERE hubspot_id IS NULL OR drive_id IS NULL OR delta_token IS NULL;
```

```
SELECT id, name, updated_at, DATEDIFF(DAY, updated_at, GETDATE()) AS
days_since_update
FROM client
WHERE updated_at < DATEADD(DAY, -30, CAST(GETDATE() AS date))
ORDER BY days_since_update DESC;
```

```
SELECT c.id, c.name, c.status,
(
    CASE WHEN c.status = 'active' THEN 3 WHEN c.status = 'prospect' THEN 2 ELSE 1 END
    + COALESCE((SELECT COUNT(*) FROM project p WHERE p.client_id = c.id), 0)
    + CASE WHEN c.hubspot_id IS NOT NULL THEN 1 ELSE 0 END
) AS activity_score
FROM client c
ORDER BY activity_score DESC
OFFSET 0 ROWS FETCH NEXT 20 ROWS ONLY;
```

```
SELECT c.industry, COUNT(DISTINCT c.id) AS total_clients, COUNT(p.id) AS total_projects
FROM client c
JOIN project p ON c.id = p.client_id
GROUP BY c.industry
```

```
HAVING COUNT(p.id) > 10
ORDER BY total_projects DESC;

SELECT owner_name,
COUNT(*) AS total_clients,
SUM(CASE WHEN status = 'active' THEN 1 ELSE 0 END) AS active_clients,
ROUND(SUM(CASE WHEN status = 'active' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS
active_ratio_pct,
MAX(created_at) AS last_client_added,
MAX(updated_at) AS last_update
FROM client
GROUP BY owner_name
ORDER BY active_ratio_pct DESC, total_clients DESC;
```

CLIENT_TEAM

```
SELECT c.name AS client_name, COUNT(ct.user_id) AS total_team_members
FROM client_team ct
JOIN client c ON ct.client_id = c.id
GROUP BY c.name
ORDER BY total_team_members DESC;

SELECT ct.role, COUNT(*) AS total_members, ROUND(AVG(ct.contracted_hours), 2) AS
avg_hours
FROM client_team ct
GROUP BY ct.role
ORDER BY total_members DESC;

SELECT u.first_name, u.last_name, SUM(ct.contracted_hours) AS total_hours,
COUNT(DISTINCT ct.client_id) AS total_clients
FROM client_team ct
JOIN [user] u ON ct.user_id = u.id
GROUP BY u.first_name, u.last_name
ORDER BY total_hours DESC;

SELECT c.name AS client_name, ct.role, COUNT(ct.user_id) AS members_in_role
FROM client_team ct
JOIN client c ON ct.client_id = c.id
GROUP BY c.name, ct.role
```

```
ORDER BY members_in_role DESC;

SELECT ct.type, COUNT(*) AS count_by_type
FROM client_team ct
GROUP BY ct.type
ORDER BY count_by_type DESC;

SELECT u.first_name, u.last_name, ct.role, ct.contracted_hours, ct.type
FROM client_team ct
JOIN [user] u ON ct.user_id = u.id
WHERE ct.type = 'external'
ORDER BY ct.contracted_hours DESC;

SELECT client_id, COUNT(*) AS num_roles
FROM client_team
GROUP BY client_id
HAVING COUNT(DISTINCT role) > 1;

SELECT ct.role, COUNT(DISTINCT ct.client_id) AS clients_covered
FROM client_team ct
GROUP BY ct.role
ORDER BY clients_covered DESC;

SELECT FORMAT(ct.created_at, 'yyyy-MM') AS month, COUNT(*) AS new_team_assignments
FROM client_team ct
GROUP BY FORMAT(ct.created_at, 'yyyy-MM')
ORDER BY month ASC;

SELECT u.first_name, u.last_name, c.name AS client_name, ct.contracted_hours, ct.role
FROM client_team ct
JOIN [user] u ON ct.user_id = u.id
JOIN client c ON ct.client_id = c.id
WHERE ct.contracted_hours BETWEEN 30 AND 45
ORDER BY ct.contracted_hours DESC;

SELECT c.name AS client_name, SUM(ct.contracted_hours) AS total_contracted_hours
FROM client_team ct
JOIN client c ON ct.client_id = c.id
GROUP BY c.name
HAVING SUM(ct.contracted_hours) > 100
ORDER BY total_contracted_hours DESC;

SELECT u.first_name, u.last_name, COUNT(DISTINCT ct.client_id) AS clients_assigned
FROM client_team ct
```

```

JOIN [user] u ON ct.user_id = u.id
GROUP BY u.first_name, u.last_name
HAVING COUNT(DISTINCT ct.client_id) > 3
ORDER BY clients_assigned DESC;

SELECT ct.role, COUNT(*) AS team_members, SUM(ct.contracted_hours) AS total_hours
FROM client_team ct
GROUP BY ct.role
ORDER BY total_hours DESC;

SELECT c.name AS client_name, u.first_name, u.last_name, ct.role, ct.type
FROM client_team ct
JOIN client c ON ct.client_id = c.id
JOIN [user] u ON ct.user_id = u.id
WHERE ct.updated_at >= DATEADD(DAY, -30, GETDATE());

```

PROJECT

```

SELECT cl.name AS client_name,COUNT(p.id) AS total_projects
FROM project p
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
ORDER BY total_projects DESC;

SELECT status,COUNT(*) AS project_count,
ROUND(COUNT(*)*100.0/(SELECT COUNT(*) FROM project),2) AS percentage
FROM project
GROUP BY status
ORDER BY project_count DESC;

SELECT category,SUM(budget) AS total_budget,SUM(amount_billed) AS total_billed,
ROUND(SUM(amount_billed)*100.0/NULLIF(SUM(budget),0),2) AS billing_progress_pct
FROM project
GROUP BY category
ORDER BY total_budget DESC;

SELECT cl.name AS client_name,SUM(p.budget) AS total_budget,SUM(p.amount_billed) AS
total_billed
FROM project p

```

```
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
HAVING SUM(p.amount_billed)>SUM(p.budget)*0.8
ORDER BY total_billed DESC;

SELECT owner_email,COUNT(*) AS owned_projects,SUM(budget) AS
total_budget,SUM(amount_billed) AS total_billed
FROM project
GROUP BY owner_email
ORDER BY owned_projects DESC;

SELECT priority,COUNT(*) AS total_projects
FROM project
GROUP BY priority
ORDER BY CASE priority WHEN 'high' THEN 1 WHEN 'medium' THEN 2 WHEN 'low' THEN 3
ELSE 4 END;

SELECT billing_type,COUNT(*) AS total_projects,ROUND(SUM(amount_billed),2) AS
total_billed
FROM project
GROUP BY billing_type
ORDER BY total_billed DESC;

SELECT currency,SUM(budget) AS total_budget,SUM(amount_billed) AS total_billed
FROM project
GROUP BY currency
ORDER BY total_billed DESC;

SELECT cl.name AS client_name,COUNT(*) AS active_projects
FROM project p
JOIN client cl ON p.client_id=cl.id
WHERE p.status='active'
GROUP BY cl.name
ORDER BY active_projects DESC;

SELECT p.name,p.status,DATEDIFF(DAY,p.start_date,p.end_date) AS project_duration_days
FROM project p
WHERE p.status IN('active','completed')
ORDER BY project_duration_days DESC;

SELECT cl.name AS client_name,AVG(p.budget_hours) AS avg_budget_hours
FROM project p
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
```

```
ORDER BY avg_budget_hours DESC;

SELECT p.name,p.amount_billed,p.budget,
ROUND(p.amount_billed*100.0/NULLIF(p.budget,0),2) AS billing_utilization_pct
FROM project p
WHERE p.status='active'
ORDER BY billing_utilization_pct DESC;

SELECT owner_email,COUNT(*) AS overdue_projects
FROM project
WHERE end_date<GETDATE() AND status<>'completed'
GROUP BY owner_email
ORDER BY overdue_projects DESC;

SELECT FORMAT(created_at,'yyyy-MM') AS month,COUNT(*) AS projects_created
FROM project
GROUP BY FORMAT(created_at,'yyyy-MM')
ORDER BY month;

SELECT cl.name AS client_name,SUM(p.amount_billed) AS total_revenue,COUNT(p.id) AS
num_projects
FROM project p
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
ORDER BY total_revenue DESC;

SELECT status,ROUND(AVG(amount_billed*100.0/NULLIF(budget,0)),2) AS
avg_billing_utilization_pct
FROM project
GROUP BY status
ORDER BY avg_billing_utilization_pct DESC;

SELECT cl.name AS client_name,COUNT(DISTINCT currency) AS currencies_used
FROM project p
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
HAVING COUNT(DISTINCT currency)>1
ORDER BY currencies_used DESC;

SELECT owner_email,MAX(updated_at) AS last_update,MIN(created_at) AS first_created
FROM project
GROUP BY owner_email
ORDER BY last_update DESC;
```

```
SELECT cl.name AS client_name,
SUM(CASE WHEN p.status='completed' THEN 1 ELSE 0 END) AS completed_projects,
SUM(CASE WHEN p.status='active' THEN 1 ELSE 0 END) AS active_projects
FROM project p
JOIN client cl ON p.client_id=cl.id
GROUP BY cl.name
ORDER BY completed_projects DESC;
```

```
SELECT category,COUNT(*) AS project_count,SUM(budget) AS total_budget
FROM project
GROUP BY category
ORDER BY total_budget DESC;
```

PROJECT_TEAM

```
SELECT p.name AS project_name,pt.title,pt.file_type,pt.file_url,pt.updated_at
FROM project_team pt
JOIN project p ON pt.project_id=p.id
ORDER BY pt.updated_at DESC;
```

```
SELECT pt.title,pt.change_type,pt.updated_at
FROM project_team pt
WHERE pt.updated_by=45
ORDER BY pt.updated_at DESC;
```

```
SELECT title,description,file_type,file_url
FROM project_team
WHERE project_id=200 AND type='document';
```

```
-- Advanced
SELECT p.name AS project_name,COUNT(pt.id) AS total_records
FROM project_team pt
JOIN project p ON pt.project_id=p.id
GROUP BY p.name
ORDER BY total_records DESC;
```

```
SELECT pt.type,COUNT(*) AS record_count,
ROUND(COUNT(*)*100.0/(SELECT COUNT(*) FROM project_team),2) AS percent_of_total
```

```
FROM project_team pt
GROUP BY pt.type
ORDER BY record_count DESC;

SELECT u.first_name,u.last_name,COUNT(pt.id) AS total_updates
FROM project_team pt
JOIN [user] u ON pt.updated_by=u.id
GROUP BY u.first_name,u.last_name
ORDER BY total_updates DESC;

SELECT p.name AS project_name,pt.type,COUNT(pt.id) AS entries_per_type
FROM project_team pt
JOIN project p ON pt.project_id=p.id
GROUP BY p.name,pt.type
ORDER BY entries_per_type DESC;

SELECT pt.project_id,MAX(pt.updated_at) AS last_update
FROM project_team pt
GROUP BY pt.project_id
ORDER BY last_update DESC;

SELECT pt.type,COUNT(DISTINCT pt.project_id) AS projects_with_type
FROM project_team pt
GROUP BY pt.type
ORDER BY projects_with_type DESC;

SELECT p.name AS project_name,COUNT(pt.id) AS total_docs
FROM project_team pt
JOIN project p ON pt.project_id=p.id
WHERE pt.type='document'
GROUP BY p.name
ORDER BY total_docs DESC;

SELECT u.first_name,u.last_name,pt.change_type,COUNT(*) AS change_count
FROM project_team pt
JOIN [user] u ON pt.updated_by=u.id
GROUP BY u.first_name,u.last_name,pt.change_type
ORDER BY change_count DESC;

SELECT pt.file_type,COUNT(*) AS file_count
FROM project_team pt
WHERE pt.file_type IS NOT NULL
GROUP BY pt.file_type
ORDER BY file_count DESC;
```

```
SELECT p.name AS project_name,pt.title,pt.type,pt.updated_at,u.first_name,u.last_name
FROM project_team pt
JOIN project p ON pt.project_id=p.id
JOIN [user] u ON pt.updated_by=u.id
WHERE pt.updated_at>=DATEADD(DAY,-30,GETDATE())
ORDER BY pt.updated_at DESC;
```

```
SELECT change_type,COUNT(*) AS changes_last_7_days
FROM project_team
WHERE updated_at>=DATEADD(DAY,-7,GETDATE())
GROUP BY change_type
ORDER BY changes_last_7_days DESC;
```

```
SELECT p.name AS project_name,
SUM(CASE WHEN pt.change_type='addition' THEN 1 ELSE 0 END) AS additions,
SUM(CASE WHEN pt.change_type='modification' THEN 1 ELSE 0 END) AS modifications,
SUM(CASE WHEN pt.change_type='deletion' THEN 1 ELSE 0 END) AS deletions
FROM project_team pt
JOIN project p ON pt.project_id=p.id
GROUP BY p.name
ORDER BY additions DESC;
```

```
SELECT u.first_name,u.last_name,COUNT(pt.id) AS total_uploaded_files
FROM project_team pt
JOIN [user] u ON pt.updated_by=u.id
WHERE pt.file_url IS NOT NULL
GROUP BY u.first_name,u.last_name
ORDER BY total_uploaded_files DESC;
```

```
SELECT p.name AS project_name,pt.title,pt.file_type,pt.file_url,pt.updated_at
FROM project_team pt
JOIN project p ON pt.project_id=p.id
WHERE pt.file_type IN('pdf','docx')
ORDER BY pt.updated_at DESC;
```

ROLE

```
SELECT r.name AS role_name,COUNT(u.id) AS total_users,  
MIN(u.created_at) AS earliest_user_created,MAX(u.updated_at) AS latest_user_update  
FROM role r
```

```
LEFT JOIN [user] u ON u.role=r.id  
GROUP BY r.name  
ORDER BY total_users DESC;
```

```
SELECT r.name AS role_name,COUNT(ct.id) AS total_client_team_members  
FROM role r  
JOIN client_team ct ON ct.role=r.name  
GROUP BY r.name  
ORDER BY total_client_team_members DESC;
```

```
SELECT r.name,COUNT(u.id) AS user_count  
FROM [user] u  
JOIN role r ON u.role=r.id  
GROUP BY r.name  
HAVING COUNT(u.id)=0;
```

```
SELECT r.name AS role_name,DATEDIFF(DAY,r.updated_at,GETDATE()) AS  
days_since_update  
FROM role r  
ORDER BY days_since_update DESC;
```

```
SELECT FORMAT(created_at,'yyyy-MM') AS month_created,COUNT(*) AS roles_created  
FROM role  
GROUP BY FORMAT(created_at,'yyyy-MM')  
ORDER BY month_created;
```

```
SELECT r.name AS role_name,COUNT(DISTINCT u.id) AS distinct_users,  
COUNT(DISTINCT ct.client_id) AS distinct_clients  
FROM role r  
LEFT JOIN [user] u ON u.role=r.id  
LEFT JOIN client_team ct ON ct.role=r.name  
GROUP BY r.name  
ORDER BY distinct_clients DESC;
```

```
SELECT r.name,COUNT(u.id) AS total_users  
FROM role r  
LEFT JOIN [user] u ON u.role=r.id  
WHERE r.name IN('Admin','Manager','Analyst')  
GROUP BY r.name  
ORDER BY total_users DESC;
```

```

SELECT r.name AS role_name,COUNT(u.id) AS assigned_users,
ROUND(COUNT(u.id)*100.0/(SELECT COUNT(*) FROM [user]),2) AS
percentage_of_total_users
FROM role r
LEFT JOIN [user] u ON u.role=r.id
GROUP BY r.name
ORDER BY percentage_of_total_users DESC;

SELECT r.name AS role_name,r.created_at,r.updated_at,
(CASE WHEN r.updated_at>DATEADD(DAY,-90,GETDATE()) THEN 'Recently Modified' ELSE
'Stale' END) AS status_flag
FROM role r
ORDER BY r.updated_at DESC;

SELECT r.name AS role_name,COUNT(u.id) AS total_users,
SUM(CASE WHEN u.employment_type='contractor' THEN 1 ELSE 0 END) AS contractors
FROM role r
LEFT JOIN [user] u ON u.role=r.id
GROUP BY r.name
ORDER BY contractors DESC;

```

USER

```

SELECT role,COUNT(*) AS total_users,
SUM(CASE WHEN employment_type='Contractor' THEN 1 ELSE 0 END) AS contractors
FROM [user]
GROUP BY role
ORDER BY total_users DESC;

SELECT employment_type,COUNT(*) AS user_count,ROUND(AVG(contractured_hour),2) AS
avg_contracted_hours
FROM [user]
GROUP BY employment_type
ORDER BY user_count DESC;

SELECT FORMAT(created_at,'yyyy-MM') AS month,COUNT(*) AS new_users
FROM [user]
GROUP BY FORMAT(created_at,'yyyy-MM')

```

```
ORDER BY month;

SELECT u.role,COUNT(ct.id) AS clients_assigned
FROM [user] u
JOIN client_team ct ON ct.user_id=u.id
GROUP BY u.role
ORDER BY clients_assigned DESC;

SELECT role,COUNT(*) AS total_users,
ROUND(COUNT(*)*100.0/(SELECT COUNT(*) FROM [user]),2) AS pct_of_total
FROM [user]
GROUP BY role
ORDER BY pct_of_total DESC;

SELECT CONCAT(first_name,' ',last_name) AS user_name,email,role,employment_type,
DATEDIFF(DAY,updated_at,GETDATE()) AS days_since_update
FROM [user]
ORDER BY days_since_update DESC;

SELECT u.role,COUNT(pt.id) AS project_updates
FROM [user] u
JOIN project_team pt ON pt.updated_by=u.id
GROUP BY u.role
ORDER BY project_updates DESC;

SELECT u.role,
SUM(CASE WHEN u.employment_type='Full-time' THEN 1 ELSE 0 END) AS full_time,
SUM(CASE WHEN u.employment_type='Contractor' THEN 1 ELSE 0 END) AS contractors
FROM [user] u
GROUP BY u.role
ORDER BY full_time DESC;

SELECT u.role,COUNT(*) AS users,MAX(updated_at) AS last_activity
FROM [user] u
GROUP BY u.role
ORDER BY last_activity DESC;

SELECT u.first_name,u.last_name,u.role,u.email,COUNT(ct.client_id) AS total_clients
FROM [user] u
JOIN client_team ct ON ct.user_id=u.id
GROUP BY u.id,u.first_name,u.last_name,u.role,u.email
ORDER BY total_clients DESC;

SELECT role,AVG(contract_hours) AS avg_contract_hours,
```

```
MAX(contract_hours) AS max_contract_hours,MIN(contract_hours) AS min_contract_hours
FROM [user]
GROUP BY role
ORDER BY avg_contract_hours DESC;

SELECT employment_type,COUNT(DISTINCT role) AS distinct_roles
FROM [user]
GROUP BY employment_type
ORDER BY distinct_roles DESC;

SELECT CONCAT(first_name,' ',last_name) AS user_name,role,created_at,updated_at,
(CASE WHEN updated_at>DATEADD(DAY,-90,GETDATE()) THEN 'Active Recently' ELSE
'Inactive' END) AS activity_flag
FROM [user]
ORDER BY updated_at DESC;

SELECT role,COUNT(*) AS total_users
FROM [user]
GROUP BY role
HAVING COUNT(*)=1
ORDER BY role;
```

XERO_CONTACTS

```
SELECT contact_status,COUNT(*) AS total_contacts
FROM xero_contacts
GROUP BY contact_status
ORDER BY total_contacts DESC;

SELECT default_currency,COUNT(contact_id) AS contacts,
SUM(CASE WHEN is_customer=1 THEN 1 ELSE 0 END) AS customers,
SUM(CASE WHEN is_supplier=1 THEN 1 ELSE 0 END) AS suppliers
FROM xero_contacts
GROUP BY default_currency
ORDER BY contacts DESC;
```

```
SELECT tracking_category_name,tracking_option_name,COUNT(*) AS contact_count
```

```
FROM xero_contacts
WHERE tracking_category_name IS NOT NULL
GROUP BY tracking_category_name,tracking_option_name
ORDER BY contact_count DESC;

SELECT is_customer,is_supplier,COUNT(*) AS total
FROM xero_contacts
GROUP BY is_customer,is_supplier
ORDER BY total DESC;

SELECT name,email_address,tax_number,default_currency
FROM xero_contacts
WHERE contact_status like '%ACTIVE%' AND (tax_number IS NULL OR tax_number="");

SELECT default_currency,COUNT(contact_id) AS total_contacts,
SUM(CASE WHEN is_customer=1 THEN 1 ELSE 0 END) AS total_customers
FROM xero_contacts
WHERE contact_status like 'ACTIVE'
GROUP BY default_currency
ORDER BY total_contacts DESC;

SELECT
name,email_address,contact_status,accounts_receivable_tax_type,accounts_payable_tax_type
FROM xero_contacts
WHERE is_supplier=1
ORDER BY name;

SELECT COUNT(*) AS active_suppliers_with_bank_details
FROM xero_contacts
WHERE is_supplier=1 AND contact_status like '%ACTIVE' AND bank_account_details IS NOT
NULL;

SELECT default_currency,COUNT(contact_id) AS total_contacts
FROM xero_contacts
WHERE is_customer=1
GROUP BY default_currency
ORDER BY total_contacts DESC;

SELECT name,email_address,contact_status,default_currency,created_at
FROM xero_contacts
WHERE is_customer=1 AND contact_status IN('ACTIVE','ARCHIVED')
ORDER BY name;

SELECT contact_status,
```

```
SUM(CASE WHEN is_customer=1 THEN 1 ELSE 0 END) AS customers,
SUM(CASE WHEN is_supplier=1 THEN 1 ELSE 0 END) AS suppliers
FROM xero_contacts
GROUP BY contact_status;

SELECT tracking_category_name,COUNT(*) AS usage_count
FROM xero_contacts
WHERE tracking_category_name IS NOT NULL
GROUP BY tracking_category_name
ORDER BY usage_count DESC;

SELECT default_currency,
SUM(CASE WHEN is_supplier=1 THEN 1 ELSE 0 END) AS total_suppliers,
SUM(CASE WHEN is_customer=1 THEN 1 ELSE 0 END) AS total_customers
FROM xero_contacts
GROUP BY default_currency
ORDER BY total_suppliers DESC;

SELECT COUNT(contact_id) AS contacts_without_email
FROM xero_contacts
WHERE email_address IS NULL OR email_address="";

SELECT contact_status,COUNT(contact_id) AS count
FROM xero_contacts
WHERE default_currency='USD'
GROUP BY contact_status
ORDER BY count DESC;
```

XERO_PROJECTS

```
SELECT status,COUNT(*) AS total_projects
FROM xero_projects
GROUP BY status
ORDER BY total_projects DESC;

SELECT currency_code,COUNT(project_id) AS project_count,SUM(total_invoiced_value) AS
total_invoiced
FROM xero_projects
GROUP BY currency_code
ORDER BY total_invoiced DESC;

SELECT contact_id,
```

```
SUM(total_task_amount_value+total_expense_amount_value) AS total_value,
SUM(total_invoiced_value) AS invoiced_value,
SUM(total_to_be_invoiced_value) AS pending_value
FROM xero_projects
GROUP BY contact_id
ORDER BY total_value DESC;

SELECT name,ROUND((total_invoiced_value*100.0/NULLIF(estimate_value,0)),2) AS
percent_billed
FROM xero_projects
WHERE estimate_value>0
ORDER BY percent_billed DESC;

SELECT contact_id,COUNT(project_id) AS active_projects
FROM xero_projects
WHERE status='INPROGRESS'
GROUP BY contact_id
ORDER BY active_projects DESC;

SELECT status,SUM(total_invoiced_value) AS total_invoiced,SUM(total_to_be_invoiced_value)
AS total_pending
FROM xero_projects
GROUP BY status;

SELECT currency_code,AVG(total_invoiced_value) AS
avg_invoiced,AVG(total_to_be_invoiced_value) AS avg_pending
FROM xero_projects
GROUP BY currency_code;

SELECT name,total_expense_amount_value,total_task_amount_value,
ROUND((total_expense_amount_value*100.0/NULLIF(total_task_amount_value,0)),2) AS
expense_ratio_pct
FROM xero_projects
WHERE total_task_amount_value>0
ORDER BY expense_ratio_pct DESC;

SELECT contact_id,COUNT(project_id) AS projects_with_deposits
FROM xero_projects
WHERE deposit_value>0
GROUP BY contact_id
ORDER BY projects_with_deposits DESC;
```

```
SELECT
name,total_invoiced_value,total_to_be_invoiced_value,total_expense_amount_value,total_task
_amount_value
FROM xero_projects
WHERE status IN('INPROGRESS','COMPLETED')
ORDER BY total_invoiced_value DESC;

SELECT name,estimate_value,total_invoiced_value,(estimate_value-total_invoiced_value) AS
variance
FROM xero_projects
WHERE estimate_value IS NOT NULL
ORDER BY variance DESC;

SELECT contact_id,COUNT(project_id) AS completed_projects,SUM(total_invoiced_value) AS
revenue_generated
FROM xero_projects
WHERE status='COMPLETED'
GROUP BY contact_id
ORDER BY revenue_generated DESC;

SELECT FORMAT(when_created,'yyyy-MM') AS month,COUNT(*) AS projects_created
FROM xero_projects
GROUP BY FORMAT(when_created,'yyyy-MM')
ORDER BY month;

SELECT name,deposit_value,deposit_applied_value,(deposit_value-deposit_applied_value) AS
unapplied_balance
FROM xero_projects
WHERE deposit_value>0
ORDER BY unapplied_balance DESC;

SELECT contact_id,
SUM(total_invoiced_value) AS total_revenue,
SUM(credit_note_amount_value) AS total_credits,
(SUM(total_invoiced_value)-SUM(credit_note_amount_value)) AS net_revenue
FROM xero_projects
GROUP BY contact_id
ORDER BY net_revenue DESC;
```