NEW YORK UNIVERSITY

School of Professional Studies

DATABASE DESIGN & MANAGEMENT

MASY1-GC 3500_1_101 | Spring 2024

Group Report Assignment #5

Final Project Report: Netflix

SUBMITTED BY: TEAM C

Pranav Borkar – BP – prb9282@nyu.edu

Saumay Killa – KS – sk10882@nyu.edu

Purva Patel – PP – pmp9467@nyu.edu

Rudra Patel – PR – rp3881@nyu.edu

Maanvi Reddy Poddutur – MR – mp6712@nyu.edu

Fengxia Yan – YF– fy2214@nyu.edu

SUBMITTED ON:

May 6th, 2024

UNDER THE GUIDANCE OF:

Prof. Amit Patel

TABLE OF CONTENTS

PROJECT SCOPE AND PLANNING DOCUMENT

Executive Summary

Netflix stands at the forefront of the streaming entertainment industry, offering a diverse range of content to millions of subscribers worldwide. Despite its success, Netflix encounters significant challenges with unauthorized password sharing, which undermines its revenue streams and compromises the integrity of user data. This pervasive issue not only affects the company's financial health but also poses risks to data security and user experience.

The core objective of this project is to develop and deploy a comprehensive database management solution specifically designed to monitor, manage, and effectively mitigate the impact of unauthorized access and password sharing. By leveraging advanced data analytics and cutting-edge technology, the project will focus on two main areas: detection of unauthorized sharing and prevention through strategic policy enforcement.

Detection: Our approach will utilize sophisticated algorithms to analyze usage patterns, identifying irregularities that suggest unauthorized access. This will involve real-time monitoring of account activity, including login locations, concurrent streams, and device usage, to flag potential breaches of Netflix's usage policies.

Prevention: Building on detection capabilities, the project will implement measures to restrict unwarranted usage promptly. This includes developing a dynamic authentication system that adapts to user behavior, implementing stricter access controls for suspicious accounts, and refining the account verification process to deter unauthorized sharing. The solution will be designed to be minimally intrusive, ensuring legitimate users experience minimal disruption.

The anticipated outcomes of this project include a significant reduction in unauthorized password sharing, enhanced security of user data, improved compliance with usage policies, and, ultimately, the protection and growth of Netflix's revenue. By addressing the challenge of password sharing head-on, Netflix will not only safeguard its business model but also reinforce its commitment to providing secure, high-quality streaming entertainment to its subscribers.

This project represents a strategic initiative to strengthen Netflix's market position and sustain its growth in the competitive streaming industry. Through innovative database solutions and a comprehensive approach to data management, we aim to set a new standard for managing and protecting digital content access in the streaming entertainment sector.

Business Case

Business Background

Netflix started in 1997, by disrupting the entertainment industry by starting a DVD rental service and shaking the traditional video rental markets. Following are some businesses ideas that Netflix implemented:

- Subscription based modeling
- Global expansion
- Creative and personalized content
- Using algorithms on databases
- Marketing and branding campaigns

Data & Management issues

Due to global expansion, large user platforms, complex algorithms, and much more, Netflix went through data and management issues. Following are some of the issues that it went through.

- Data Issue: As Netflix has a large database of the users, their personalized content, information, likings, history, etc. The quality of the data is compromised giving rise to problems like data quality, data governance, data scaling, data security, data compliance, data integrity and handling the personalized content, data storage, data analysis etc.
- Management Issue: Managing a large database can be a tedious task. Managing data includes data security, managing large volumes of data, handling passwords of the accounts, sharing credentials, OLTP-Online transaction processing systems, data analysis for a user, managing number of devices, IP tracking, restrictions on password sharing - also one of the issues it is facing right now, subscriber retention and acquisition.

Keeping all of these in mind, Netflix has launched a privacy policy where it will not allow password sharing on multiple devices and will be limited to only members within the same household to reduce the risk of data breach and video piracy.

Business Impact

Password sharing is one of the major issues that Netflix went through, for the last few years before it found the solution. 85% of people in America share their password among each other. Due to this password sharing and video piracy Netflix reported losing almost $190 million every month which is highest among all the online streaming platforms (Demers, 2023).

Database solution

Keeping this in mind Netflix launched a policy of not sharing passwords other than members within the same household. OLTP along with RDBMS can be used as a solution to overcome the issue of password sharing and video piracy.

- OLTP verification: The Online transaction processing system provides real time data of the users, associated profiles, and personalized content. Using OLTP whenever a user logs in, the address can be quickly verified with the address in the database for checking the associated devices.
- User location identification: Similarly using OLTP the user location can be checked in reference to the allowed locations on the database for that user, if there is an unauthorized location detected the OLTP can deny it automatically and notify the user.
- User Identification in real time: If OLTP is linked with RDBMS, the location of the user can be verified on the database whenever the user logs in keeping their privacy intact. Based on this user can also know if there is unauthorized access to their account.
- Real time tie up with ISPs: Using OLTP, Netflix can tie up with the ISPs in order to check for the traffic of user logins on a single account, that helps with knowing the shared devices, profiles or accounts.

Benefit to business using Database Solution

The database solution leveraging OLTP and RDBMS offers Netflix a comprehensive and effective strategy for addressing the challenges of password sharing, enhancing security, improving user experience, and safeguarding revenue streams.

- Enhanced Security and Accountability: By implementing real-time user verification and location identification using OLTP and RDBMS, Netflix can enhance security measures and hold users accountable for their account usage. This helps to prevent unauthorized access and ensures that only legitimate users within the same household have access to the service, reducing the risk of password sharing and piracy.
- Improved User Experience: Users benefit from a more personalized and secure experience with the ability to identify unauthorized access in real-time. By linking OLTP with RDBMS, Netflix can provide users with insights into their account activity, including device usage and login locations, empowering them to detect and address any security concerns promptly.
- Efficient Management of Access Controls: The integration of OLTP and RDBMS allows for efficient management of access controls based on user location and device association. Netflix can automatically deny access from unauthorized locations and devices, reducing the administrative burden and ensuring compliance with content licensing agreements.
- Proactive Detection of Unauthorized Sharing: Real-time tie-ups with ISPs enable Netflix to monitor traffic patterns and detect instances of password sharing or account misuse promptly. This proactive approach allows Netflix to take immediate action to address unauthorized access and enforce account policies, mitigating the impact of piracy on revenue and content distribution.
- Streamlined Collaboration with ISPs: The use of OLTP facilitates real-time collaboration with ISPs for traffic analysis and enforcement efforts. By partnering with ISPs, Netflix can leverage their network infrastructure to identify and block unauthorized access, creating a unified approach to combating password sharing and piracy.

Project Objective

Develop and deploy a Netflix Password Protection Feature within four months, enhancing account security, user control, and reducing unauthorized access incidents, while continuously monitoring and refining based on user feedback with the objective to bring down the 85% of password sharing rate to approximately 20% along with improving the overall revenue.

Scope Statement

Project scope

Database Development Life Cycle for Netflix Password Sharing Protection Feature:

- Planning
    - Define requirements for the database system, including data storage, access control, and security measures.
    - Conduct feasibility studies to assess the technical and financial viability of implementing the password sharing protection feature.
    - Identify stakeholders and establish communication channels for gathering input and feedback.
    - Develop a project plan outlining timelines, milestones, and resource requirements for database development.
- Analysis
    - Gather detailed requirements for the database schema, including tables, fields, and relationships.
    - Analyze existing data structures and systems to determine compatibility and integration requirements.
    - Identify potential security risks and compliance requirements related to password sharing and user authentication.
    - Define user roles and access levels to enforce security measures and prevent unauthorized access to sensitive data.
- Design
    - Design the database schema based on the requirements gathered in the analysis phase, ensuring normalization and data integrity.
    - Define data access and retrieval mechanisms, including queries, views, and stored procedures.
    - Design authentication and authorization mechanisms to control access to the password sharing feature and user account data.
    - Develop a data encryption strategy to protect sensitive information, such as user credentials and viewing history.
- Implementation
    - Create the database schema using appropriate database management systems (e.g., MySQL, PostgreSQL).
    - Implement data access and retrieval logic using SQL queries and database programming languages (e.g., SQL, PL/SQL).
    - Develop authentication and authorization modules to validate user credentials and enforce access controls.
    - Integrate encryption algorithms and security protocols to safeguard user data during transmission and storage.
- Testing
    - Conduct unit tests to validate individual database components, including tables, queries, and stored procedures.

    - Perform integration tests to ensure seamless interaction between the database and application layers.
    - Execute security tests to assess vulnerabilities and validate the effectiveness of security measures.
    - Conduct performance tests to evaluate database scalability and responsiveness under various load conditions.
- Deployment
    - Deploy the database system to production environments, following best practices for deployment and configuration.
    - Perform data migration and population tasks to transfer existing user data and configurations to the new system.
    - Configure monitoring and logging mechanisms to track database performance and security events.
    - Conduct user training and documentation to educate stakeholders on how to use the password sharing protection feature effectively.
- Maintenance
    - Monitor database performance and security metrics to identify potential issues and proactively address them.
    - Apply patches and updates to database software and security protocols to mitigate vulnerabilities.
    - Perform regular backups and disaster recovery procedures to protect against data loss and system downtime.
    - Collect user feedback and feature requests to inform future enhancements and improvements to the password sharing protection feature.

Product Scope

To ensure the security and functionality of the Netflix Password Sharing Protection feature, the database system should incorporate various features. Here are some essential features:

- User Authentication and Authorization
    - Implement robust authentication mechanisms to ensure that only authorized users can access the database.
    - Utilize role-based access control (RBAC) to assign appropriate permissions to users based on their roles and responsibilities within the system.
- Encryption and Data Protection
    - Employ encryption techniques to protect sensitive data such as user credentials, viewing history, and payment information.
    - Use hashing algorithms to securely store passwords and other sensitive information in the database.
- Activity Logging and Auditing
    - Log all database activities, including user logins, access attempts, and data modifications, to maintain an audit trail for security and compliance purposes.
    - Implement mechanisms for reviewing and analyzing audit logs to detect and investigate suspicious activities.
- Access Control Policies

    - Define and enforce access control policies to restrict access to sensitive data based on user roles, privileges, and contextual factors such as time and location.
    - Implement granular access controls to ensure that users can only access the data necessary for their authorized tasks.
- Data Integrity and Consistency
    - Implement measures to ensure the integrity and consistency of data stored in the database, such as constraints, validations, and referential integrity rules.
    - Use transactions and locking mechanisms to maintain data consistency during concurrent access and updates.
- Backup and Recovery
    - Implement regular backups of the database to protect against data loss due to hardware failures, natural disasters, or other unforeseen events.
    - Develop a robust disaster recovery plan to minimize downtime and restore data in case of a catastrophic failure.
- Scalability and Performance Optimization
    - Design the database system to scale horizontally and vertically to accommodate growing data volumes and user loads.
    - Optimize database performance through indexing, query optimization, and caching strategies to ensure responsive and efficient access to data.
- Compliance and Regulatory Requirements
    - Ensure that the database system complies with relevant regulatory requirements such as GDPR, CCPA, and industry-specific standards for data protection and privacy.
    - Implement features for data anonymization, pseudonymization, and user consent management to facilitate compliance with privacy regulations.
- Monitoring and Alerting
    - Implement monitoring tools and alerts to proactively detect and respond to performance issues, security threats, and potential breaches in real-time.
    - Set up automated alerts for abnormal database activities, unauthorized access attempts, and critical system events.
- High Availability and Fault Tolerance
    - Design the database system for high availability and fault tolerance to minimize downtime and ensure continuous access to data.
    - Implement redundancy, failover mechanisms, and load balancing strategies to mitigate the impact of hardware failures and network outages.

Assumptions and Constraints

Assumptions

- Users are willing to adopt and utilize the password protection feature.
- The feature will effectively deter unauthorized access without hindering legitimate user sharing practices.
- Users have access to internet connectivity and compatible devices for accessing the feature.
- The implementation of robust security measures will sufficiently protect user data from breaches and cyber threats.
- Throughout the deployment phase, the project will be able to constantly monitor and enhance the password security functionality depending on user feedback.

Constraints

- Compliance with legal and regulatory frameworks related to data privacy and security.
- Integration with existing Netflix infrastructure and systems without causing service disruptions.
- Resource constraints including budget, time, and personnel for development, testing, and maintenance.
- Compatibility with a wide range of devices and operating systems used by Netflix subscribers.
- Balancing security measures with user convenience to ensure a seamless and user-friendly experience.
- Managing user expectations and potential resistance to changes in account sharing practices.

Project Team

Team Members	Roles	Email
Pranav Borkar	Software Developer	prb9282@nyu.edu
Saumay Killa	Data Analyst	sk10882@nyu.edu
Purva Patel	Business Analyst	pmp9467@nyu.edu
Rudra Patel	Project Manager	rp3881@nyu.edu
Maanvi Reddy	Risk Analyst	mp6712@nyu.edu

Major Deliverables, Due Dates, & Acceptance Criteria

Phase	Major Deliverable	Due Date	Acceptance Criteria
Requirements Analysis	Requirements Specification Document	02/25/2024	• Document includes detailed requirements gathered. • Requirements are clear, unambiguous, and traceable.
Conceptual Design	Entity-Relationship Diagram (ERD)	03/11/2024	• ERD represents the high-level data model. • Entities, relationships, and attributes are accurately defined. • The model aligns with business requirements.
Logical Design	Normalized Data Model	03/25/2024	• Data model is in at least 3rd normal form (3NF). • All relationships are properly defined. • Attributes and data types are specified.
Physical Design	Database Schema	04/06/2024	• Physical database schema is derived from the logical model. • Tables, indexes, and constraints are defined. • Considerations for performance and scalability are documented.
Implementation	Deployed Database	04/20/2024	• Database is successfully implemented in the production environment. • Data migration is executed without errors. • Security measures are implemented as per requirements.

Testing	Test Plan and Test Cases	05/04/2024	• Test plan covers unit, integration, and system testing. • Test cases are created, executed, and passed successfully. • Any identified bugs or issues are documented and resolved.
Deployment	Deployment Documentation	05/20/2024	• Deployment plan is executed without major issues. • Rollback plan is in place and tested. • Monitoring and backup procedures are established.
Maintenance and Optimization	Maintenance Plan	Ongoing through every phase	• Plan for ongoing maintenance and update is documented. • Performance monitoring tools are in place. • Continuous improvement plan is established. • Overall Project Acceptance Criteria • All major deliverables from each phase meet the specified acceptance criteria. • The database solution meets performance and scalability requirements.

DATA MODEL

Logical Model

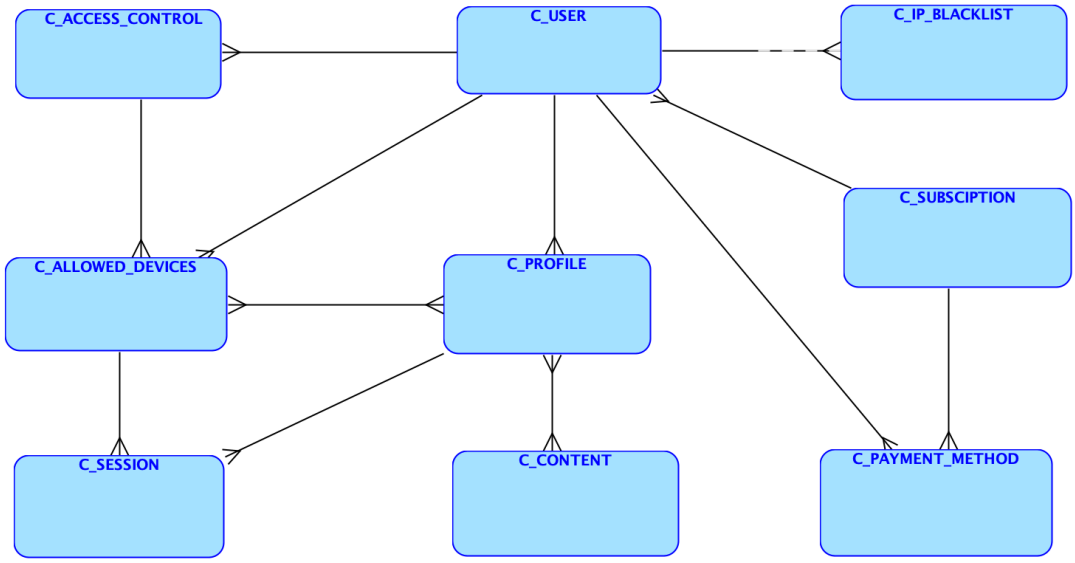


Figure 1 Enterprise Model

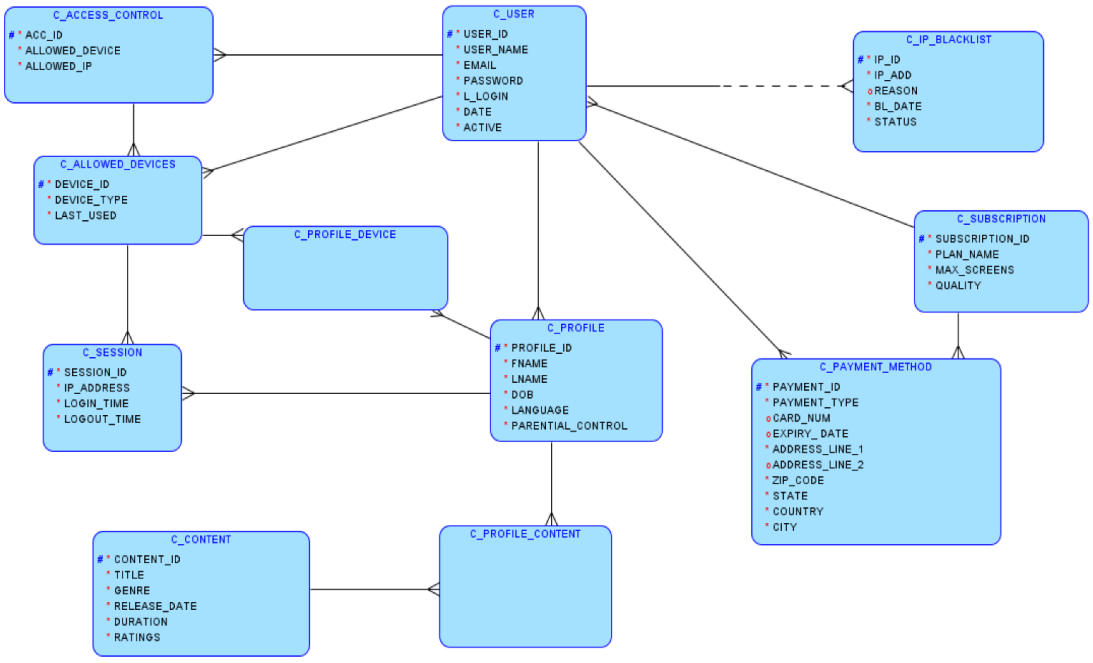


Figure 2 Logical Model

Relational Model

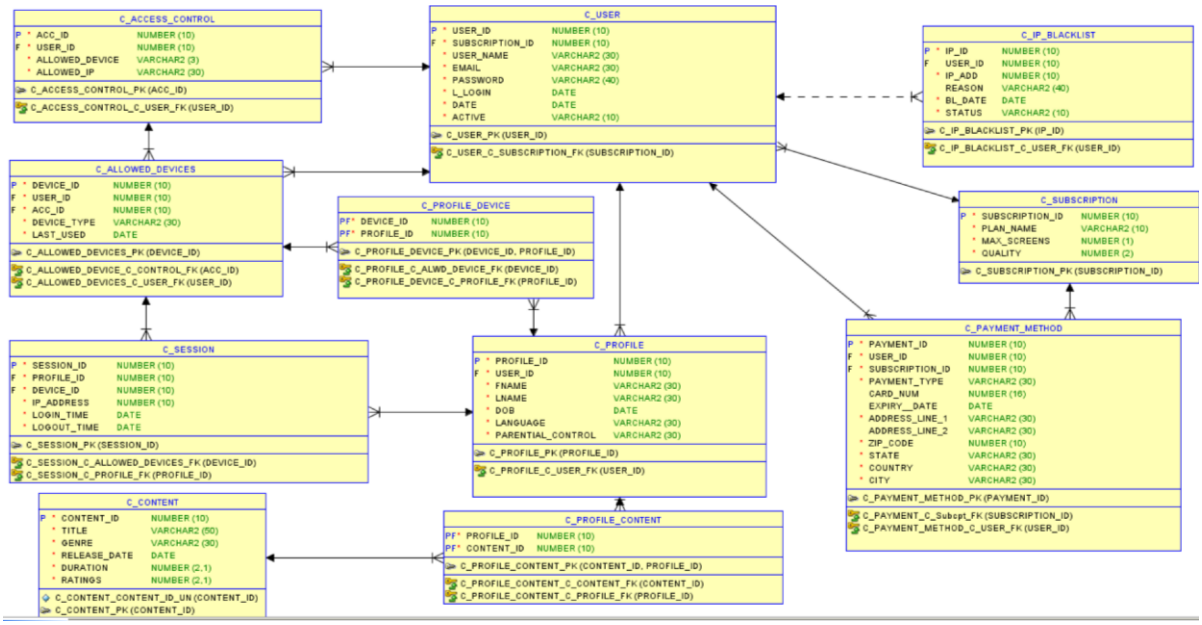


Figure 3 Relational Model

Model Assumptions

In the relational database model for the streaming service, every user is distinguished by a unique identifier and has the ability to register multiple allowed devices as well as create various personalized profiles. These profiles contain specific settings, such as language preferences and parental controls, ensuring content is tailored and appropriate for each viewer. User interactions with the service are tracked through sessions, which log the device used and the IP address from which the service was accessed. Content, the core component of the service, is accessible by all profiles and is enhanced with attributes to aid in searching and providing recommendations. Allowed devices must be registered to individual users, and they serve as the portals through which profiles access available content.

Moreover, subscription plans are crucial as they outline the services provided to the user, including the number of screens that can be used simultaneously and the quality of the content available. Payment methods are tied to user accounts to handle billing, and various options are offered, ranging from credit cards to digital payment systems like PayPal, allowing users to select their preferred payment methods for subscriptions and other transactions. Security is a significant concern; thus, IP blacklists are actively managed to block access from specific addresses that may pose security threats or are flagged for suspicious activities. Similarly, access controls are established, designating which allowed devices and IP addresses are authorized for service access, providing an additional layer of security, and ensuring compliance with regional content distribution laws.

- Profiles to Allowed devices: Many-to-Many: Profiles can use multiple allowed devices and vice versa; this is managed via a bridge entity called Profile_Device.
- User to Profile: One-to-Many: A single user can have multiple profiles, but each profile is associated with one user.
- User to Allowed Device: One-to-Many: A user can register multiple allowed devices to their account, but each allowed device is specific to one user.
- User to IP Blacklist: One-to-Many (Optional): A user may have multiple IP addresses blacklisted. The dashed line indicates that this is an optional relationship since not all users' IP addresses will be blocked, suggesting that specific actions trigger an IP being blacklisted for a user.
- User to Access Control: One-to-Many: Each user could have multiple devices and hence the multiple access control settings but each access control setting would be applicable to a single device hence a single user.
- User to Payment Method: One-to-Many: Users can have multiple payment methods on file.
- User to Subscription: Many-to-One: Each user will subscribe to one subscription plan but each subscription plan can be subscribed by multiple users.
- Profile to Content: Many-to-Many: A profile can access multiple content pieces, and each content piece can be accessed by multiple profiles; this is managed via a bridge entity called Profile_Content.
- Profile to Session: One-to-Many: A profile can have multiple sessions, indicating each login and logout activity.
- Session to Allowed Device: Many-to-One: A session is associated with a single device, but a device can have multiple sessions over time.

- Payment Methods to Subscriptions: Many-to-One: The subscription plan can be made through different payment methods, but each payment will go towards one subscription.

Each assumption aligns with a goal to create a secure, user-friendly, and robust system for managing streaming services. The models aim to capture the multifaceted relationships between users, their preferences, allowed devices, content, and security measures. The presence of entities like Access Control and IP Blacklist indicates a strong emphasis on security and maintaining control over who can access the service and how.

DDL Code

```
-- Generated by Oracle SQL Developer Data Modeler
23.1.0.087.0806
--   at:       2024-03-09 13:43:01 EST
--   site:     Oracle Database 21c
--   type:     Oracle Database 21c


-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE CREATE TABLE
c_access_control (
acc_id         NUMBER(10) NOT NULL,
user_id        NUMBER(10) NOT NULL,
allowed_devices VARCHAR2(3) NOT NULL, allowed_ips
     VARCHAR2(30) NOT NULL
);


COMMENT ON COLUMN c_access_control.acc_id IS 'Unique control
access ID ';

COMMENT ON COLUMN c_access_control.allowed_devices IS 'Number of
the devices user account is allowed on';

COMMENT ON COLUMN c_access_control.allowed_ips IS 'Allowed IPs
';
CREATE UNIQUE INDEX c_access_control

idx ON c_access_control (user_id ASC );


ALTER TABLE c_access_control ADD CONSTRAINT c_access_control_pk
PRIMARY KEY ( acc_id );

CREATE TABLE c_content (
content_id   NUMBER(10) NOT NULL, profile_id   NUMBER(10) NOT
NULL, title         VARCHAR2(50) NOT NULL,
genre     VARCHAR2(30) NOT NULL,
release_date DATE NOT NULL, duration    NUMBER(2, 1) NOT NULL,
ratings   NUMBER(2, 1) NOT NULL
);

COMMENT ON COLUMN c_content.content_id IS 'Unique Content ID';

COMMENT ON COLUMN c_content.title IS 'Ttile of the content';
```

```
COMMENT ON COLUMN c_content.genre IS 'Type of content';


COMMENT ON COLUMN c_content.release_date IS 'Release date of the
content ';

COMMENT ON COLUMN c_content.duration IS 'Duration of the
content';

COMMENT ON COLUMN c_content.ratings IS 'Ratings of the content';

ALTER TABLE c_content ADD CONSTRAINT c_content_pk PRIMARY KEY (
content_id
);


CREATE TABLE c_device (
device_id           NUMBER(10) NOT NULL, user_id  NUMBER(10) NOT
NULL,
device_type VARCHAR2(30) NOT NULL, last_used   DATE NOT NULL
);

COMMENT ON COLUMN c_device.device_id IS 'Unique ID of the
device';


COMMENT ON COLUMN c_device.last_used IS 'Last used date of the
device';
ALTER TABLE c_device ADD CONSTRAINT c_device_pk PRIMARY KEY (
device_id ); CREATE TABLE c_ip_blacklist (
ip_add NUMBER(10) NOT NULL,
user_id NUMBER(10),

reason  VARCHAR2(40), bl_date DATE NOT NULL, status VARCHAR2(10)
NOT NULL
);


COMMENT ON COLUMN c_ip_blacklist.ip_add IS 'IP Address of the
unwanted devices ';


COMMENT ON COLUMN c_ip_blacklist.reason IS 'Reason why the
device is blacklisted ';
```

```
COMMENT ON COLUMN c_ip_blacklist.bl_date IS 'Date of black
list';


COMMENT ON COLUMN c_ip_blacklist.status IS 'Status of the IP
address';


ALTER TABLE c_ip_blacklist ADD CONSTRAINT c_ip_blacklist_pk
PRIMARY KEY ( ip_add );


CREATE TABLE c_payment_method ( paymentid    NUMBER(10) NOT
NULL, user_id       NUMBER(10) NOT NULL,
payment_type   VARCHAR2(30) NOT NULL, card_num
     NUMBER(16),
expiry_date    DATE, subscription_id NUMBER(10) NOT NULL,
bill_add       VARCHAR2(30) NOT NULL,
zip_code       NUMBER(10) NOT NULL,
state          VARCHAR2(30) NOT NULL
);

COMMENT ON COLUMN c_payment_method.paymentid IS 'Unique ID of
the payment method';


COMMENT ON COLUMN c_payment_method.payment_type IS 'Type of the
payment method';


COMMENT ON COLUMN c_payment_method.card_num IS 'Number of the
card';


COMMENT ON COLUMN c_payment_method.bill_add IS 'Billing address
of the user ';

COMMENT ON COLUMN c_payment_method.zip_code IS 'Zip code of the
address';

COMMENT ON COLUMN c_payment_method.state IS 'State of the
billing address';

ALTER TABLE c_payment_method ADD CONSTRAINT c_payment_method_pk
PRIMARY KEY ( paymentid );

CREATE TABLE c_profile (
```

```
profile_id     NUMBER(10) NOT NULL,
user_id        NUMBER(10) NOT NULL,
name           VARCHAR2(30) NOT NULL,
age            NUMBER(2) NOT NULL,
language       VARCHAR2(30) NOT NULL,
parential_control VARCHAR2(30) NOT NULL
);

COMMENT ON COLUMN c_profile.profile_id IS 'Unique Profile ID ';

COMMENT ON COLUMN c_profile.name IS 'Name of the Profile';

COMMENT ON COLUMN c_profile.age IS 'Age based restricted
contents';

COMMENT ON COLUMN c_profile.language IS 'Preferred language ';

COMMENT ON COLUMN c_profile.parential_control IS 'Kids profile
control ';

ALTER TABLE c_profile ADD CONSTRAINT c_profile_pk PRIMARY KEY (
profile_id
);

CREATE TABLE c_profile_device ( device_id NUMBER(10) NOT NULL,
profile_id NUMBER(10) NOT NULL
);

ALTER TABLE c_profile_device ADD CONSTRAINT c_profile_device_pk
PRIMARY KEY ( profile_id,
device_id );

CREATE TABLE c_session (
session_id          NUMBER(10) NOT NULL,
profile_id          NUMBER(10) NOT NULL,
ip_address          VARCHAR2(30) NOT NULL,
logintime           DATE NOT NULL,
logouttime          DATE NOT NULL, c_device_device_id NUMBER(10)
NOT NULL
);


COMMENT ON COLUMN c_session.session_id IS 'Unique session ID';

COMMENT ON COLUMN c_session.ip_address IS 'IP address of the
device';
```

```
COMMENT ON COLUMN c_session.logintime IS 'Profile Login Time';


COMMENT ON COLUMN c_session.logouttime IS 'User logout time';


ALTER TABLE c_session ADD CONSTRAINT c_session_pk PRIMARY KEY (
session_id
);

CREATE TABLE c_subscription ( subscription_id NUMBER(10) NOT
NULL, plan_name     VARCHAR2(10) NOT NULL, max_screens
     NUMBER(1) NOT NULL, quality        NUMBER(2) NOT NULL
);


COMMENT ON COLUMN c_subscription.subscription_id IS 'Unique
subcription ID';


COMMENT ON COLUMN c_subscription.plan_name IS 'Name of the
associated user plan';

COMMENT ON COLUMN c_subscription.max_screens IS 'Maximum allowed
screens';

COMMENT ON COLUMN c_subscription.quality IS
'The quality of the plan based on the subscription plan';


ALTER TABLE c_subscription ADD CONSTRAINT c_subscription_pk
PRIMARY KEY ( subscription_id );

CREATE TABLE c_user (
user_id        NUMBER(10) NOT NULL,
user_name      VARCHAR2(30) NOT NULL, email
     VARCHAR2(30) NOT NULL,
password       VARCHAR2(40) NOT NULL,
l_login        DATE NOT NULL,
"DATE"         DATE NOT NULL,
active         VARCHAR2(10) NOT NULL,
subscription_id NUMBER(10) NOT NULL
);

COMMENT ON COLUMN c_user.user_id IS 'Unique User ID for each
user';
```

```
COMMENT ON COLUMN c_user.user_name IS 'Name of the user';

COMMENT ON COLUMN c_user.email IS 'Email id of user';

COMMENT ON COLUMN c_user.password IS 'Password of each user
account';

COMMENT ON COLUMN c_user.l_login IS 'Last Login date and time';

COMMENT ON COLUMN c_user."DATE" IS
'Creation Date of account';

COMMENT ON COLUMN c_user.active IS 'Activity Status ';

ALTER TABLE c_user ADD CONSTRAINT c_user_pk PRIMARY KEY (
user_id );
ALTER TABLE c_access_control
ADD CONSTRAINT c_access_control_c_user_fk FOREIGN KEY ( user_id
) REFERENCES c_user ( user_id );

ALTER TABLE c_content
ADD CONSTRAINT c_content_c_profile_fk FOREIGN KEY ( profile_id )
REFERENCES c_profile ( profile_id );

ALTER TABLE c_device
ADD CONSTRAINT c_device_c_user_fk FOREIGN KEY ( user_id )
REFERENCES c_user ( user_id );


ALTER TABLE c_ip_blacklist
ADD CONSTRAINT c_ip_blacklist_c_user_fk FOREIGN KEY ( user_id )
REFERENCES c_user ( user_id );


ALTER TABLE c_payment_method
ADD CONSTRAINT c_pay_met_c_user_fk FOREIGN KEY ( user_id )
REFERENCES c_user ( user_id );

ALTER TABLE c_payment_method
ADD CONSTRAINT c_payment_met_c_subs_fk FOREIGN KEY (
subscription_id ) REFERENCES c_subscription ( subscription_id );

ALTER TABLE c_profile_device
ADD CONSTRAINT c_pro_dev_c_pro_fk FOREIGN KEY ( profile_id )
REFERENCES c_profile ( profile_id );
```

```
ALTER TABLE c_profile
ADD CONSTRAINT c_profile_c_user_fk FOREIGN KEY ( user_id )
REFERENCES c_user ( user_id );


ALTER TABLE c_profile_device
ADD CONSTRAINT c_profile_device_c_device_fk FOREIGN KEY (
device_id ) REFERENCES c_device ( device_id );


ALTER TABLE c_session
ADD CONSTRAINT c_session_c_device_fk FOREIGN KEY (
c_device_device_id ) REFERENCES c_device ( device_id );

ALTER TABLE c_session

ADD CONSTRAINT c_session_c_profile_fk FOREIGN KEY ( profile_id )
REFERENCES c_profile ( profile_id );

ALTER TABLE c_user
ADD CONSTRAINT c_user_c_subscription_fk FOREIGN KEY (
subscription_id ) REFERENCES c_subscription ( subscription_id );


--   Oracle SQL Developer Data Modeler  Summary Report:
--
--   CREATE TABLE   10
--   CREATE INDEX   1
--   ALTER TABLE    22
--   CREATE VIEW    0
--   ALTER VIEW     0
--   CREATE PACKAGE 0
--   CREATE PACKAGE BODY 0
--   CREATE PROCEDURE    0
--   CREATE FUNCTION     0
--   CREATE TRIGGER 0
--   ALTER TRIGGER  0
--   CREATE COLLECTION TYPE   0
--   CREATE STRUCTURED TYPE   0
--   CREATE STRUCTURED TYPE BODY   0
--   CREATE CLUSTER 0
--   CREATE CONTEXT 0
--   CREATE DATABASE     0
--   CREATE DIMENSION    0
--   CREATE DIRECTORY    0
--   CREATE DISK GROUP   0
--   CREATE ROLE    0
```

```
--   CREATE ROLLBACK SEGMENT  0
--   CREATE SEQUENCE     0
--   CREATE MATERIALIZED VIEW 0
--   CREATE MATERIALIZED VIEW LOG  0
--   CREATE SYNONYM 0
--   CREATE TABLESPACE   0
--   CREATE USER    0
--
--   DROP TABLESPACE     0
--   DROP DATABASE  0
--
--   REDACTION POLICY    0
--
--   ORDS DROP SCHEMA    0
--   ORDS ENABLE SCHEMA  0
--   ORDS ENABLE OBJECT  0
--
--   ERRORS    0
--   WARNINGS  0
```

DATA DICTIONARY QUERIES RESULT SCREENSHOTS

List of Tables

select table_name
from user_tables;

```
1  select table_name from user_tables
```

TABLE_NAME
C_ACCESS_CONTROL
C_ALLOWED_DEVICES
C_CONTENT
C_IP_BLACKLIST
C_PAYMENT_METHOD
C_PROFILE
C_PROFILE_CONTENT
C_PROFILE_DEVICE
C_SESSION
C_SUBSCRIPTION
C_USER

Download CSV

List of Table Columns

```
select table_name, column_name,column_id
from user_tab_columns
order by table_name,column_id;
```

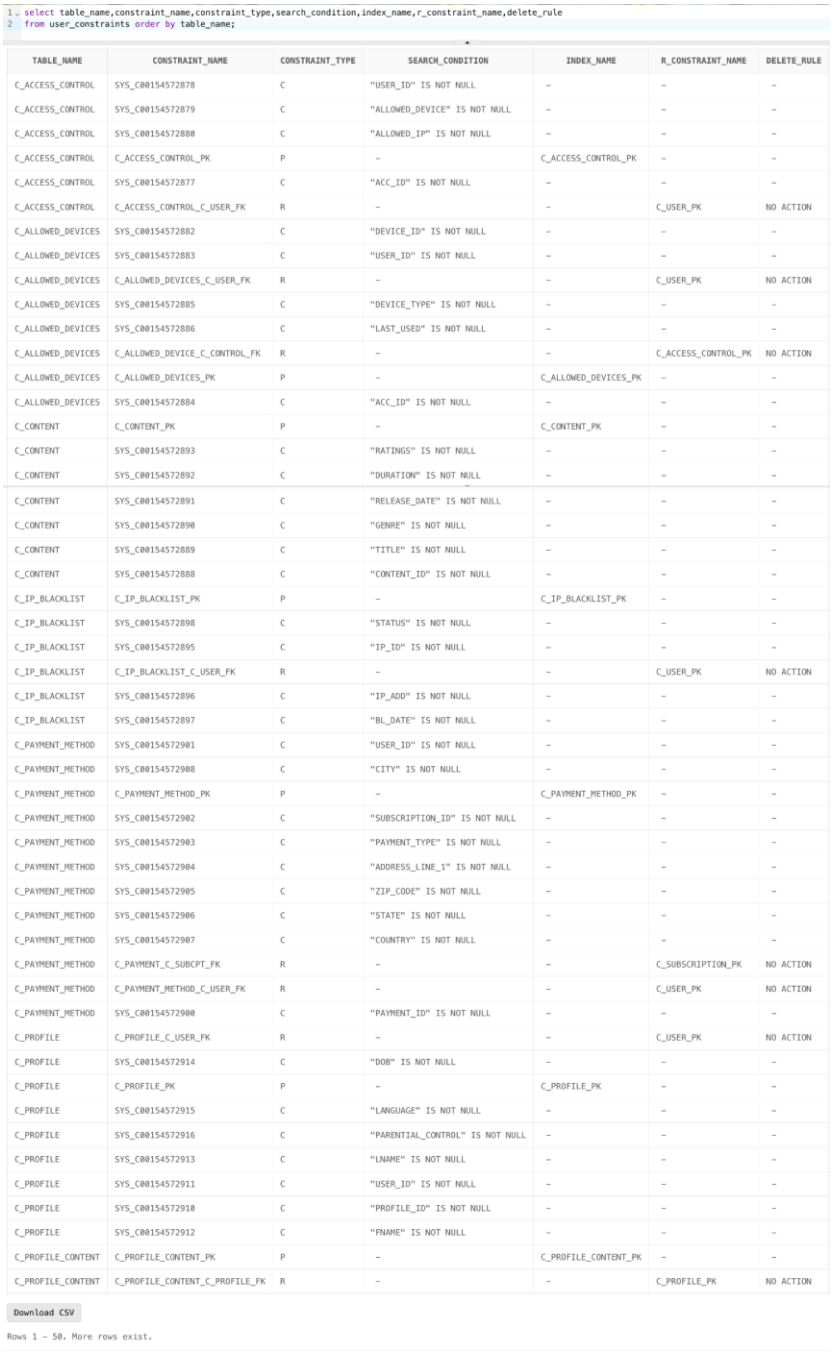
```
1 select table_name, column_name,column_id from user_tab_columns order by table_name,column_id;
```

TABLE_NAME	COLUMN_NAME	COLUMN_ID
C_ACCESS_CONTROL	ACC_ID	1
C_ACCESS_CONTROL	USER_ID	2
C_ACCESS_CONTROL	ALLOWED_DEVICE	3
C_ACCESS_CONTROL	ALLOWED_IP	4
C_ALLOWED_DEVICES	DEVICE_ID	1
C_ALLOWED_DEVICES	USER_ID	2
C_ALLOWED_DEVICES	ACC_ID	3
C_ALLOWED_DEVICES	DEVICE_TYPE	4
C_ALLOWED_DEVICES	LAST_USED	5
C_CONTENT	CONTENT_ID	1
C_CONTENT	TITLE	2
C_CONTENT	GENRE	3
C_CONTENT	RELEASE_DATE	4
C_CONTENT	DURATION	5
C_CONTENT	RATINGS	6
C_IP_BLACKLIST	IP_ID	1
C_IP_BLACKLIST	USER_ID	2
C_IP_BLACKLIST	IP_ADD	3
C_IP_BLACKLIST	REASON	4
C_IP_BLACKLIST	BL_DATE	5
C_IP_BLACKLIST	STATUS	6
C_PAYMENT_METHOD	PAYMENT_ID	1
C_PAYMENT_METHOD	USER_ID	2
C_PAYMENT_METHOD	SUBSCRIPTION_ID	3
C_PAYMENT_METHOD	PAYMENT_TYPE	4
C_PAYMENT_METHOD	CARD_NUM	5
C_PAYMENT_METHOD	EXPIRY__DATE	6
C_PAYMENT_METHOD	ADDRESS_LINE_1	7
C_PAYMENT_METHOD	ADDRESS_LINE_2	8
C_PAYMENT_METHOD	ZIP_CODE	9
C_PAYMENT_METHOD	STATE	10
C_PAYMENT_METHOD	COUNTRY	11
C_PAYMENT_METHOD	CITY	12
C_PROFILE	PROFILE_ID	1
C_PROFILE	USER_ID	2
C_PROFILE	FNAME	3
C_PROFILE	LNAME	4
C_PROFILE	DOB	5
C_PROFILE	LANGUAGE	6
C_PROFILE	PARENTIAL_CONTROL	7
C_PROFILE_CONTENT	PROFILE_ID	1
C_PROFILE_CONTENT	CONTENT_ID	2
C_PROFILE_DEVICE	DEVICE_ID	1
C_PROFILE_DEVICE	PROFILE_ID	2
C_SESSION	SESSION_ID	1
C_SESSION	PROFILE_ID	2
C_SESSION	DEVICE_ID	3
C_SESSION	IP_ADDRESS	4
C_SESSION	LOGIN_TIME	5
C_SESSION	LOGOUT_TIME	6

Download CSV

List of Table Column Constraints

```
select
table_name,constraint_name,constraint_type,search_condition,index_name,r_constraint_name,delete_rule
from user_constraints
order by table_name;
```



```
1 select table_name,constraint_name,constraint_type,search_condition,index_name,r_constraint_name,delete_rule
2 from user_constraints order by table_name;
```

TABLE_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	INDEX_NAME	R_CONSTRAINT_NAME	DELETE_RULE
C_ACCESS_CONTROL	SYS_C00154572878	C	"USER_ID" IS NOT NULL	–	–	–
C_ACCESS_CONTROL	SYS_C00154572879	C	"ALLOWED_DEVICE" IS NOT NULL	–	–	–
C_ACCESS_CONTROL	SYS_C00154572880	C	"ALLOWED_IP" IS NOT NULL	–	–	–
C_ACCESS_CONTROL	C_ACCESS_CONTROL_PK	P	–	C_ACCESS_CONTROL_PK	–	–
C_ACCESS_CONTROL	SYS_C00154572877	C	"ACC_ID" IS NOT NULL	–	–	–
C_ACCESS_CONTROL	C_ACCESS_CONTROL_C_USER_FK	R	–	–	C_USER_PK	NO ACTION
C_ALLOWED_DEVICES	SYS_C00154572882	C	"DEVICE_ID" IS NOT NULL	–	–	–
C_ALLOWED_DEVICES	SYS_C00154572883	C	"USER_ID" IS NOT NULL	–	–	–
C_ALLOWED_DEVICES	C_ALLOWED_DEVICES_C_USER_FK	R	–	–	C_USER_PK	NO ACTION
C_ALLOWED_DEVICES	SYS_C00154572885	C	"DEVICE_TYPE" IS NOT NULL	–	–	–
C_ALLOWED_DEVICES	SYS_C00154572886	C	"LAST_USED" IS NOT NULL	–	–	–
C_ALLOWED_DEVICES	C_ALLOWED_DEVICE_C_CONTROL_FK	R	–	–	C_ACCESS_CONTROL_PK	NO ACTION
C_ALLOWED_DEVICES	C_ALLOWED_DEVICES_PK	P	–	C_ALLOWED_DEVICES_PK	–	–
C_ALLOWED_DEVICES	SYS_C00154572884	C	"ACC_ID" IS NOT NULL	–	–	–
C_CONTENT	C_CONTENT_PK	P	–	C_CONTENT_PK	–	–
C_CONTENT	SYS_C00154572893	C	"RATINGS" IS NOT NULL	–	–	–
C_CONTENT	SYS_C00154572892	C	"DURATION" IS NOT NULL	–	–	–
C_CONTENT	SYS_C00154572891	C	"RELEASE_DATE" IS NOT NULL	–	–	–
C_CONTENT	SYS_C00154572890	C	"GENRE" IS NOT NULL	–	–	–
C_CONTENT	SYS_C00154572889	C	"TITLE" IS NOT NULL	–	–	–
C_CONTENT	SYS_C00154572888	C	"CONTENT_ID" IS NOT NULL	–	–	–
C_IP_BLACKLIST	C_IP_BLACKLIST_PK	P	–	C_IP_BLACKLIST_PK	–	–
C_IP_BLACKLIST	SYS_C00154572898	C	"STATUS" IS NOT NULL	–	–	–
C_IP_BLACKLIST	SYS_C00154572895	C	"IP_ID" IS NOT NULL	–	–	–
C_IP_BLACKLIST	C_IP_BLACKLIST_C_USER_FK	R	–	–	C_USER_PK	NO ACTION
C_IP_BLACKLIST	SYS_C00154572896	C	"IP_ADD" IS NOT NULL	–	–	–
C_IP_BLACKLIST	SYS_C00154572897	C	"BL_DATE" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572901	C	"USER_ID" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572908	C	"CITY" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	C_PAYMENT_METHOD_PK	P	–	C_PAYMENT_METHOD_PK	–	–
C_PAYMENT_METHOD	SYS_C00154572902	C	"SUBSCRIPTION_ID" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572903	C	"PAYMENT_TYPE" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572904	C	"ADDRESS_LINE_1" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572905	C	"ZIP_CODE" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572906	C	"STATE" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	SYS_C00154572907	C	"COUNTRY" IS NOT NULL	–	–	–
C_PAYMENT_METHOD	C_PAYMENT_C_SUBCPT_FK	R	–	–	C_SUBSCRIPTION_PK	NO ACTION
C_PAYMENT_METHOD	C_PAYMENT_METHOD_C_USER_FK	R	–	–	C_USER_PK	NO ACTION
C_PAYMENT_METHOD	SYS_C00154572900	C	"PAYMENT_ID" IS NOT NULL	–	–	–
C_PROFILE	C_PROFILE_C_USER_FK	R	–	–	C_USER_PK	NO ACTION
C_PROFILE	SYS_C00154572914	C	"DOB" IS NOT NULL	–	–	–
C_PROFILE	C_PROFILE_PK	P	–	C_PROFILE_PK	–	–
C_PROFILE	SYS_C00154572915	C	"LANGUAGE" IS NOT NULL	–	–	–
C_PROFILE	SYS_C00154572916	C	"PARENTIAL_CONTROL" IS NOT NULL	–	–	–
C_PROFILE	SYS_C00154572913	C	"LNAME" IS NOT NULL	–	–	–
C_PROFILE	SYS_C00154572911	C	"USER_ID" IS NOT NULL	–	–	–
C_PROFILE	SYS_C00154572910	C	"PROFILE_ID" IS NOT NULL	–	–	–
C_PROFILE	SYS_C00154572912	C	"FNAME" IS NOT NULL	–	–	–
C_PROFILE_CONTENT	C_PROFILE_CONTENT_PK	P	–	C_PROFILE_CONTENT_PK	–	–
C_PROFILE_CONTENT	C_PROFILE_CONTENT_C_PROFILE_FK	R	–	–	C_PROFILE_PK	NO ACTION

Download CSV

Rows 1 – 50. More rows exist.

List of Table Column Comments

```
select table_name,column_name,comments
from user_col_comments
order by table_name;
```

```
1 select table_name,column_name,comments from user_col_comments order by table_name;
```

TABLE_NAME	COLUMN_NAME	COMMENTS
C_ACCESS_CONTROL	USER_ID	-
C_ACCESS_CONTROL	ALLOWED_DEVICE	Number of the devices user account is allowed on
C_ACCESS_CONTROL	ALLOWED_IP	Allowed IPs
C_ACCESS_CONTROL	ACC_ID	Unique control access ID
C_ALLOWED_DEVICES	USER_ID	-
C_ALLOWED_DEVICES	ACC_ID	-
C_ALLOWED_DEVICES	DEVICE_TYPE	-
C_ALLOWED_DEVICES	LAST_USED	Last used date of the device
C_ALLOWED_DEVICES	DEVICE_ID	Unique ID of the device
C_CONTENT	CONTENT_ID	Unique Content ID
C_CONTENT	TITLE	Ttile of the content
C_CONTENT	RATINGS	Ratings of the content
C_CONTENT	DURATION	Duration of the content
C_CONTENT	RELEASE_DATE	Release date of the content
C_CONTENT	GENRE	Type of content
C_IP_BLACKLIST	IP_ID	Unique IP ID
C_IP_BLACKLIST	BL_DATE	Date of black list
C_IP_BLACKLIST	REASON	Reason why the device is blacklisted
C_IP_BLACKLIST	IP_ADD	IP Address of the unwanted devices
C_IP_BLACKLIST	USER_ID	-
C_IP_BLACKLIST	STATUS	Status of the IP address
C_PAYMENT_METHOD	ZIP_CODE	Zip code of the address
C_PAYMENT_METHOD	ADDRESS_LINE_2	Address Line 2/Landmark(s) if any
C_PAYMENT_METHOD	ADDRESS_LINE_1	Street address of the user
C_PAYMENT_METHOD	EXPIRY__DATE	Expiry date of card
C_PAYMENT_METHOD	CARD_NUM	Number of the card
C_PAYMENT_METHOD	PAYMENT_TYPE	Type of the payment method
C_PAYMENT_METHOD	SUBSCRIPTION_ID	-
C_PAYMENT_METHOD	USER_ID	-
C_PAYMENT_METHOD	PAYMENT_ID	Unique ID of the payment method
C_PAYMENT_METHOD	STATE	State of the billing address
C_PAYMENT_METHOD	CITY	City of adress provided
C_PAYMENT_METHOD	COUNTRY	Country of adress provided
C_PROFILE	FNAME	First Name of the profile
C_PROFILE	USER_ID	-
C_PROFILE	LNAME	Last name of the profile
C_PROFILE	DOB	Date of birth of the user for restricted content
C_PROFILE	LANGUAGE	-
C_PROFILE	PARENTIAL_CONTROL	Control restrictions for kids
C_PROFILE_CONTENT	PROFILE_ID	-
C_PROFILE_CONTENT	CONTENT_ID	-
C_PROFILE_DEVICE	DEVICE_ID	-
C_PROFILE_DEVICE	PROFILE_ID	-
C_SESSION	DEVICE_ID	-
C_SESSION	PROFILE_ID	-
C_SESSION	SESSION_ID	Uniques Session ID
C_SESSION	IP_ADDRESS	IP address of the session device/server.
C_SESSION	LOGOUT_TIME	Logout time of the session
C_SESSION	LOGIN_TIME	Log in time of the session

Download CSV

Rows 1 - 50. More rows exist.

NUMBER OF RECORDS & POLUATED DATA

Total Number of Records for Each Table

User Table

```
1  SELECT COUNT(*) AS "USER TABLE" FROM C_USER;
```

USER TABLE
14

Download CSV

Profile Table

```
1  select count(*) as "Profile Table" from C_PROFILE
```

Profile Table
13

Download CSV

Payment Method Table

```
1 select count(*) as "Payment Method Table" from C_PAYMENT_METHOD
```

Payment Method Table
12

Download CSV

Content Table

```
1 SELECT COUNT(*) AS "CONTENT TABLE" FROM C_CONTENT
```

CONTENT TABLE
15

Download CSV

Access Control Table

```
1 select count(*) as "Access Control Table" from C_ACCESS_CONTROL
```

Access Control Table
13

Download CSV

Subscription Table

```
1 select count(*) as "Subscription Table" from C_SUBSCRIPTION
```

Subscription Table
9

Download CSV

IP Blacklist Table

```
1 Select count(*) as "IP BLACKLIST TABLE" from C_IP_BLACKLIST
```

IP BLACKLIST TABLE
12

Download CSV

Allowed Devices Table

```
1 SELECT COUNT(*) AS "ALLOWED DEVICES TABLE" from C_ALLOWED_DEVICES
```

ALLOWED DEVICES TABLE
13

Download CSV

Session Table

```
1  select count(*) as "SESSION TABLE" from C_SESSION
```

SESSION TABLE
13

Download CSV

Profile Device Table

```
1  SELECT COUNT(*) AS "PROFILE DEVICE TABLE " from C_PROFILE_DEVICE
```

PROFILE DEVICE TABLE
20

Download CSV

Profile Content Table

```
1  SELECT COUNT(*) AS "PROFILE CONTENT" FROM C_PROFILE_CONTENT
```

PROFILE CONTENT
20

Download CSV

Populated Data for Each Table

User Table

```
SELECT * FROM C_USER
```

USER_ID	SUBSCRIPTION_ID	USER_NAME	EMAIL	PASSWORD	L_LOGIN	DATE	ACTIVE
1001	2001	Rudra	rp@nyu.edu	123	20-APR-24	06-MAY-24	Yes
1002	2002	Purva	pp@nyu.edu	321	21-MAY-24	06-MAY-24	No
1003	2003	Maanvi	mr@nyu.edu	12343	22-JUN-24	06-MAY-24	Yes
1004	2001	Saumay	sk@nyu.edu	1234321	23-APR-24	06-MAY-24	No
1005	2005	Pranav	pb@nyu.edu	123213	20-APR-24	06-MAY-24	Yes
1006	2006	Dhoni	msd@nyu.edu	1243213	25-APR-24	06-MAY-24	No
1007	2008	Sachin	st@nyu.edu	12ccds3	26-APR-24	06-MAY-24	Yes
1008	2004	Rohit	rs@nyu.edu	123423	30-APR-24	06-MAY-24	No
1009	2003	Virat	vk@nyu.edu	12dsa3	21-APR-24	06-MAY-24	Yes
1010	2002	Amit	ap@nyu.edu	12dsa3	22-APR-24	06-MAY-24	No
1011	2001	Charlie	cc@nyu.edu	12dsdaa3	12-MAY-24	06-MAY-24	No
1012	2005	Chaplin	cp@nyu.edu	12ddasa3	23-MAY-24	06-MAY-24	No
1013	2009	User	us@nyu.edu	12dfdsasa3	21-MAY-24	06-MAY-24	No
1014	2009	Admin	am@nyu.edu	12dsa3	10-MAY-24	06-MAY-24	No

Download CSV

Subscription Table

```
SELECT * FROM C_SUBSCRIPTION
```

SUBSCRIPTION_ID	PLAN_NAME	MAX_SCREENS	QUALITY
2001	Basic	1	SD
2002	Standard	2	HD
2003	Premium	4	4K
2004	Ultra	6	8K
2005	Family	4	HD
2006	Lite	1	SD
2007	Starter	1	SD
2008	Plus	2	HD
2009	Profession	4	4K

Download CSV

Session Table

```
1 SELECT * FROM C_SESSION
```

SESSION_ID	PROFILE_ID	DEVICE_ID	IP_ADDRESS	LOGIN_TIME	LOGOUT_TIME
1	1	1	3232235877	15-APR-24	15-APR-24
2	2	2	3232235878	16-APR-24	16-APR-24
3	3	3	3232235879	17-APR-24	17-APR-24
4	4	4	3232235880	18-APR-24	18-APR-24
5	5	5	3232235881	19-APR-24	19-APR-24
6	6	6	3232235882	20-APR-24	20-APR-24
7	7	7	3232235883	21-APR-24	21-APR-24
8	8	8	3232235884	22-APR-24	22-APR-24
9	9	9	3232235885	23-APR-24	23-APR-24
10	10	10	3232235886	24-APR-24	24-APR-24
11	11	11	3232235887	25-APR-24	25-APR-24
12	12	12	3232235888	26-APR-24	26-APR-24
13	13	13	3232235889	27-APR-24	27-APR-24

Download CSV

13 rows selected.

Profile Table

```
1 select * from C_PROFILE
2
```

PROFILE_ID	USER_ID	FNAME	LNAME	DOB	LANGUAGE	PARENTIAL_CONTROL
1	1001	Rudra	Patel	18-OCT-01	English	No
2	1002	Purva	Patel	17-OCT-01	English	No
3	1003	Maanvi	Reddy	12-OCT-01	English	No
4	1004	Saumay	Killa	23-FEB-01	English	No
5	1005	Pranav	Latecome	10-DEC-00	English	No
6	1006	Dhoni	MS	18-OCT-01	English	No
7	1007	Sachin	Tendulkar	20-OCT-01	English	No
8	1008	Rohit	Sharma	21-OCT-01	English	No
9	1009	Virat	Kohli	22-OCT-01	English	No
10	1010	Amit	Patel	23-OCT-01	English	No
11	1011	Charlie	Chaplin	24-OCT-01	English	No
12	1012	Chaplin	Chalie	25-OCT-01	English	No
13	1013	Ben	Taylor	26-OCT-01	English	No

Download CSV

13 rows selected.

Profile Device Table

```
1  SELECT * FROM C_PROFILE_DEVICE
```

DEVICE_ID	PROFILE_ID
1	1
1	3
1	9
2	2
2	3
3	2
3	6
3	8
4	4
4	5
5	4
5	7
5	8
7	6
8	1
8	2
9	1
9	2
9	3
9	4

Download CSV

20 rows selected.

Profile Content Table

```
1  SELECT * FROM C_PROFILE_CONTENT
```

PROFILE_ID	CONTENT_ID
1	1
3	1
5	1
6	1
2	2
3	2
7	2
3	3
6	3
7	3
1	4
4	4
5	4
8	4
2	5
6	5
8	5
2	6
3	6
5	6

Download CSV

20 rows selected.

Payment Method Table

```
1 SELECT * FROM C_PAYMENT_METHOD
```

PAYMENT_ID	USER_ID	SUBSCRIPTION_ID	PAYMENT_TYPE	CARD_NUM	EXPIRY__DATE	ADDRESS_LINE_1	ADDRESS_LINE_2	ZIP_CODE	STATE	COUNTRY	CITY
1	1001	2001	Credit Card	1234567890123456	15-APR-24	123 Main St	Apt 1A	12345	NY	USA	New York
2	1002	2002	Debit Card	9876543210987654	15-APR-24	456 Elm St	Apt 2B	54321	CA	USA	Los Angeles
3	1003	2003	PayPal	6543210987654321	15-APR-24	789 Oak St	Apt 3C	67890	TX	USA	Houston
4	1004	2001	Credit Card	1230123012301230	15-APR-24	101 Pine St	Apt 4D	54321	CA	USA	Los Angeles
5	1005	2005	Credit Card	4567890123456789	15-APR-24	555 Cedar St	Apt 5E	34567	WA	USA	Seattle
6	1006	2006	PayPal	2222333344445555	15-APR-24	666 Elm St	Apt 6F	67890	TX	USA	Houston
7	1007	2008	Debit Card	8888999911112222	15-APR-24	777 Maple St	Apt 7G	23456	FL	USA	Miami
8	1008	2004	Credit Card	9999999900000000	15-APR-24	888 Birch St	Apt 8H	45678	CA	USA	Los Angeles
9	1009	2003	Credit Card	7777888899990000	15-APR-24	999 Walnut St	Apt 9I	56789	NY	USA	New York
10	1010	2002	PayPal	1234123412341234	15-APR-24	111 Pine St	Apt 10J	67890	TX	USA	Houston
11	1011	2001	Debit Card	4567456745674567	15-APR-24	222 Oak St	Apt 11K	34567	WA	USA	Seattle
12	1012	2005	Credit Card	9876987698769876	15-APR-24	444 Cedar St	Apt 12L	45678	CA	USA	Los Angeles

Download CSV

12 rows selected.

IP Blacklist Table

```
1 SELECT * FROM C_IP_BLACKLIST
```

IP_ID	USER_ID	IP_ADD	REASON	BL_DATE	STATUS
1	1001	19216811	Too many failed login attempts	15-APR-24	Blacklist
2	1002	19216812	Suspicious activity detected	16-APR-24	Blacklist
3	1003	19216813	Reported as spam	17-APR-24	Blacklist
4	1004	19216814	Malicious activity detected	18-APR-24	Blacklist
5	1005	19216815	Unauthorized access attempt	19-APR-24	Blacklist
6	1006	19216816	Previously reported as compromised	20-APR-24	Blacklist
7	1007	19216817	Suspected malware infection	21-APR-24	Blacklist
8	1008	19216818	Known proxy server	22-APR-24	Blacklist
9	1009	19216819	Phishing activity detected	23-APR-24	Blacklist
10	1010	192168110	Flagged by intrusion detection system	24-APR-24	Blacklist
11	1011	192168111	Reported as part of botnet	25-APR-24	Blacklist
12	1012	192168112	malicious servers	26-APR-24	Blacklist

Download CSV

12 rows selected.

Content Table

```
1  SELECT * FROM C_CONTENT
```

CONTENT_ID	TITLE	GENRE	RELEASE_DATE	DURATION	RATINGS
1	Stranger Things	Sci-Fi	15-JUL-16	6	8.7
2	The Crown	Drama	04-NOV-16	6	8.7
3	Money Heist	Crime	02-MAY-17	5	8.3
4	The Witcher	Fantasy	20-DEC-19	6	8.2
5	Narcos	Crime	28-AUG-15	5	8.8
6	Breaking Bad	Drama	20-JAN-08	4.5	9.5
7	Black Mirror	Sci-Fi	04-DEC-11	6	8.8
8	Friends	Comedy	22-SEP-94	2.2	8.9
9	The Office	Comedy	24-MAR-05	2.2	8.9
10	Stranger Things 2	Sci-Fi	27-OCT-17	6	8.7
11	Breaking Bad: El Camino	Drama	11-OCT-19	1.2	7.3
12	Dark	Sci-Fi	01-DEC-17	6	8.8
13	BoJack Horseman	Animation	22-AUG-14	2.5	8.7
14	Orange Is the New Black	Comedy-Drama	11-JUL-13	5	8.1
15	Mindhunter	Crime-Drama	13-OCT-17	5	8.6

Download CSV

15 rows selected.

Allowed Devices Table

```
1  SELECT * FROM C_ALLOWED_DEVICES
```

DEVICE_ID	USER_ID	ACC_ID	DEVICE_TYPE	LAST_USED
1	1001	1	Desktop	15-APR-24
2	1002	2	Mobile	16-APR-24
3	1003	3	Tablet	17-APR-24
4	1004	4	Desktop	18-APR-24
5	1005	5	Mobile	19-APR-24
6	1006	6	Desktop	20-APR-24
7	1007	7	Mobile	21-APR-24
8	1008	8	Desktop	22-APR-24
9	1009	9	Tablet	23-APR-24
10	1010	10	Mobile	24-APR-24
11	1011	11	Desktop	25-APR-24
12	1012	12	Mobile	26-APR-24
13	1013	13	Tablet	27-APR-24

Download CSV

13 rows selected.

Access Control Table

```
1 SELECT * FROM C_ACCESS_CONTROL
```

ACC_ID	USER_ID	ALLOWED_DEVICE	ALLOWED_IP
1	1001	1	192.168.1.101
2	1002	2	192.168.1.102
3	1003	1	192.168.1.103
4	1004	1	192.168.1.104
5	1005	2	192.168.1.105
6	1006	1	192.168.1.106
7	1007	1	192.168.1.107
8	1008	2	192.168.1.108
9	1009	2	192.168.1.109
10	1010	1	192.168.1.110
11	1011	1	192.168.1.111
12	1012	1	192.168.1.112
13	1013	1	192.168.1.113

Download CSV

13 rows selected.

SQL QUERIES FOR INFORMATION ANALYSIS

Two queries using Subqueries

Query 1

```
1  Select Content_ID AS "CONTENT ID",
2         TITLE AS "MOVIE NAME",
3         RATINGS AS "RATINGS"
4  FROM C_CONTENT C1
5  WHERE RATINGS > (SELECT AVG(C2.RATINGS) FROM C_CONTENT C2 WHERE C2.GENRE=C1.GENRE)
```

CONTENT ID	MOVIE NAME	RATINGS
2	The Crown	8.7
5	Narcos	8.8
6	Breaking Bad	9.5
7	Black Mirror	8.8
12	Dark	8.8

Download CSV

5 rows selected.

Business Purpose of the Query

The query helps identify content (e.g., movies, shows, etc.) with a higher-than-average rating within its genre. This is useful for businesses to:

- Spot popular content within each genre.
- Enhance recommendations for users.
- Optimize content strategy to focus on successful content.

Explanation of the Query

The query selects the CONTENT_ID, TITLE, and RATINGS from the C_CONTENT table where the rating (RATINGS) of each piece of content is greater than the average rating in the same genre (GENRE).

- SELECT CONTENT_ID, TITLE, RATINGS: Specifies the columns to select from the C_CONTENT table.
- FROM C_CONTENT c1: Specifies the C_CONTENT table and aliases it as c1.
- WHERE RATINGS > (: Filters content based on the following subquery.
- SELECT AVG(c2.RATINGS): Subquery calculates the average rating (AVG(c2.RATINGS)) of content within the same genre.

- FROM C_CONTENT c2: Uses the C_CONTENT table and aliases it as c2.
- WHERE c2.GENRE = c1.GENRE: Filters the subquery results to only include content in the same genre.

This query selects content with above-average ratings within each genre, helping businesses understand which content is most successful.

Query 2

```
Select USER_NAME AS "USER NAME",
        EMAIL AS "USER EMAIL"
FROM C_USER
WHERE USER_ID IN (SELECT USER_ID FROM C_IP_BLACKLIST  WHERE STATUS = 'Blacklist');
```

USER NAME	USER EMAIL
Rudra	rp@nyu.edu
Purva	pp@nyu.edu
Maanvi	mr@nyu.edu
Saumay	sk@nyu.edu
Pranav	pb@nyu.edu
Dhoni	msd@nyu.edu
Sachin	st@nyu.edu
Rohit	rs@nyu.edu
Virat	vk@nyu.edu
Amit	ap@nyu.edu
Charlie	cc@nyu.edu
Chaplin	cp@nyu.edu

Download CSV

12 rows selected.

Business Purpose of the Query

The query is used to identify users who have been blacklisted due to security or other business reasons. By retrieving the names and email addresses of these users, a business can manage its blacklist effectively, taking further actions such as monitoring, communicating with the users, or revoking their access if necessary.

Explanation of the Query

The query retrieves the USER_NAME and EMAIL columns from the C_USER table for users who have been blacklisted. This is determined by checking the user IDs in a blacklist table (C_IP_BLACKLIST).

- SELECT USER_NAME, EMAIL: Specifies the columns to retrieve from the C_USER table.
- FROM C_USER: Specifies the C_USER table from which to retrieve the data.
- WHERE USER_ID IN (: Filters the C_USER table to only include rows where the USER_ID is present in the results of the following subquery.
- SELECT USER_ID: Subquery selects the USER_ID column from the C_IP_BLACKLIST table.
- FROM C_IP_BLACKLIST: Specifies the table C_IP_BLACKLIST from which to retrieve the data.
- WHERE STATUS = 'Blacklist': Filters the subquery results to only include user IDs that are blacklisted (i.e., where STATUS equals 'Blacklist').

In summary, this query identifies and retrieves the names and email addresses of users whose user IDs are present in the C_IP_BLACKLIST table and who have a status of 'Blacklist.'

Two queries using Table joins (minimum three table joins)

Query 1

```
Select PAYMENT_TYPE AS "Payment Method",
CARD_NUM AS "CARD NUMBER",
ADDRESS_LINE_1 || ' ' || ADDRESS_LINE_2 || ', ' || A.STATE ||  ', ' || COUNTRY || ', ' || ZIP_CODE AS "Address",
PLAN_NAME AS "Plan Name",
QUALITY AS "Quality Allowed",
USER_NAME AS "User Name",
EMAIL AS "User Email",
Active AS "Status of Subscription"
from C_PAYMENT_METHOD A  JOIN C_SUBSCRIPTION USING (SUBSCRIPTION_ID) JOIN C_USER USING (USER_ID);
```

Payment Method	CARD NUMBER	Address	Plan Name	Quality Allowed	User Name	User Email	Status of Subscription
Credit Card	1234567890123456	123 Main St Apt 1A, NY, USA, 12345	Basic	SD	Rudra	rp@nyu.edu	Yes
Debit Card	9876543210987654	456 Elm St Apt 2B, CA, USA, 54321	Standard	HD	Purva	pp@nyu.edu	No
PayPal	6543210987654321	789 Oak St Apt 3C, TX, USA, 67890	Premium	4K	Maanvi	mr@nyu.edu	Yes
Credit Card	1230123012301230	101 Pine St Apt 4D, CA, USA, 54321	Basic	SD	Saumay	sk@nyu.edu	No
Credit Card	4567890123456789	555 Cedar St Apt 5E, WA, USA, 34567	Family	HD	Pranav	pb@nyu.edu	Yes
PayPal	2222333344445555	666 Elm St Apt 6F, TX, USA, 67890	Lite	SD	Dhoni	msd@nyu.edu	No
Debit Card	8888999911112222	777 Maple St Apt 7G, FL, USA, 23456	Plus	HD	Sachin	st@nyu.edu	Yes
Credit Card	9999999900000000	888 Birch St Apt 8H, CA, USA, 45678	Ultra	8K	Rohit	rs@nyu.edu	No
Credit Card	7777888899990000	999 Walnut St Apt 9I, NY, USA, 56789	Premium	4K	Virat	vk@nyu.edu	Yes
PayPal	1234123412341234	111 Pine St Apt 10J, TX, USA, 67890	Standard	HD	Amit	ap@nyu.edu	No
Debit Card	4567456745674567	222 Oak St Apt 11K, WA, USA, 34567	Basic	SD	Charlie	cc@nyu.edu	No
Credit Card	9876987698769876	444 Cedar St Apt 12L, CA, USA, 45678	Family	HD	Chaplin	cp@nyu.edu	No

Download CSV

12 rows selected.

Business Purpose of the Query

The query is used to identify the relationships between users, their subscriptions, and payment methods. This information is useful for businesses to understand how users are managing their subscriptions and payments, allowing them to tailor services and manage customer accounts effectively.

Explanation of the Query

The query joins three tables: C_PAYMENT_METHOD, C_SUBSCRIPTION, and C_USER, using common columns (SUBSCRIPTION_ID and USER_ID) to retrieve all records from these tables. This enables the business to view the combined data of a user's subscription and payment method, as well as the corresponding user details.

- SELECT: This keyword indicates that the following columns will be selected and returned in the result set.
- PAYMENT_TYPE AS "Payment Method": This renames the column PAYMENT_TYPE to "Payment Method" in the result set.
- CARD_NUM AS "CARD NUMBER": This renames the column CARD_NUM to "CARD NUMBER" in the result set.
- ADDRESS_LINE_1 || ' ' || ADDRESS_LINE_2 || ', ' || A.STATE || ', ' || COUNTRY || ', ' || ZIP_CODE AS "Address": This concatenates several columns (ADDRESS_LINE_1, ADDRESS_LINE_2, A.STATE, COUNTRY, and ZIP_CODE) together to form a

complete address, separated by commas and spaces. It renames the result to "Address" in the result set.

- PLAN_NAME AS "Plan Name": This renames the column PLAN_NAME to "Plan Name" in the result set.
- QUALITY AS "Quality Allowed": This renames the column QUALITY to "Quality Allowed" in the result set.
- USER_NAME AS "User Name": This renames the column USER_NAME to "User Name" in the result set.
- EMAIL AS "User Email": This renames the column EMAIL to "User Email" in the result set.
- Active AS "Status of Subscription": This renames the column Active to "Status of Subscription" in the result set.
- FROM C_PAYMENT_METHOD: Specifies the C_PAYMENT_METHOD table as the primary table.
- JOIN C_SUBSCRIPTION USING (SUBSCRIPTION_ID): Joins the C_SUBSCRIPTION table using the SUBSCRIPTION_ID column, allowing the query to match records with the same SUBSCRIPTION_ID.
- JOIN C_USER USING (USER_ID): Joins the C_USER table using the USER_ID column, allowing the query to match records with the same USER_ID.

Query 2

```
select
    USER_NAME AS "USER NAME",
    EMAIL AS "USER EMAIL",
    ACTIVE AS "STATUS OF SUBSCRIPTION",
    ALLOWED_IP AS "IP ALLOWED",
    DEVICE_TYPE AS "DEVICE TYPE",
    LAST_USED "LAST USED DATE"
from C_ALLOWED_DEVICES JOIN C_USER USING (USER_ID) JOIN C_ACCESS_CONTROL USING (ACC_ID)
```

USER NAME	USER EMAIL	STATUS OF SUBSCRIPTION	IP ALLOWED	DEVICE TYPE	LAST USED DATE
Rudra	rp@nyu.edu	Yes	192.168.1.101	Desktop	15-APR-24
Purva	pp@nyu.edu	No	192.168.1.102	Mobile	16-APR-24
Maanvi	mr@nyu.edu	Yes	192.168.1.103	Tablet	17-APR-24
Saumay	sk@nyu.edu	No	192.168.1.104	Desktop	18-APR-24
Pranav	pb@nyu.edu	Yes	192.168.1.105	Mobile	19-APR-24
Dhoni	msd@nyu.edu	No	192.168.1.106	Desktop	20-APR-24
Sachin	st@nyu.edu	Yes	192.168.1.107	Mobile	21-APR-24
Rohit	rs@nyu.edu	No	192.168.1.108	Desktop	22-APR-24
Virat	vk@nyu.edu	Yes	192.168.1.109	Tablet	23-APR-24
Amit	ap@nyu.edu	No	192.168.1.110	Mobile	24-APR-24
Charlie	cc@nyu.edu	No	192.168.1.111	Desktop	25-APR-24
Chaplin	cp@nyu.edu	No	192.168.1.112	Mobile	26-APR-24
User	us@nyu.edu	No	192.168.1.113	Tablet	27-APR-24

Download CSV

13 rows selected.

Business Purpose of the Query

Query is designed to help identify users who have been blacklisted due to security or other business reasons. By retrieving the names and email addresses of these users, Netflix can effectively manage their blacklist and take appropriate actions such as monitoring, communicating with the users, or revoking their access if necessary.

Explanation of the Query

- SELECT: This keyword indicates that the following columns will be selected and returned in the result set.
- USER_NAME AS "USER NAME": This renames the column USER_NAME to "USER NAME" in the result set.
- EMAIL AS "USER EMAIL": This renames the column EMAIL to "USER EMAIL" in the result set.
- ACTIVE AS "STATUS OF SUBSCRIPTION": This renames the column ACTIVE to "STATUS OF SUBSCRIPTION" in the result set.
- ALLOWED_IP AS "IP ALLOWED": This renames the column ALLOWED_IP to "IP ALLOWED" in the result set.
- DEVICE_TYPE AS "DEVICE TYPE": This renames the column DEVICE_TYPE to "DEVICE TYPE" in the result set.
- LAST_USED "LAST USED DATE": This renames the column LAST_USED to "LAST USED DATE" in the result set.
- C_ALLOWED_DEVICE: This table likely contains information about devices that are allowed for users. In the context of Amazon, this could be devices like smartphones, tablets, or computers that users can use to access Netflix.
- C_USER: This table contains user information such as user IDs, names, emails, etc. In the context of Amazon, this would represent registered users of the platform.
- C_ACCESS: This table seems to be involved in granting access rights, but without the join condition, it's unclear what specific purpose it serves. Assuming it contains information about user access permissions or levels.

One query using in-line View

```
SELECT
    p.PROFILE_ID as "Profile ID",
    p.FNAME || ' ' || p.LNAME AS "User Name",
    p.USER_ID as "User ID",
    u_stats.TOTAL_PROFILES as "Number of Profiles",
    u_stats.TOTAL_CONTENT as "Number of Contents",
    TRUNC(u_stats.AVG_RATINGS, 2) AS "Average Rating",
    u_stats.MIN_RATINGS as "Min Ratings",
    u_stats.MAX_RATINGS as "Max Ratings"
FROM
    C_PROFILE p,
    (SELECT
        pr.USER_ID,
        COUNT(DISTINCT pr.PROFILE_ID) AS TOTAL_PROFILES,
        COUNT(pc.CONTENT_ID) AS TOTAL_CONTENT,
        AVG(c.RATINGS) AS AVG_RATINGS,
        MIN(c.RATINGS) AS MIN_RATINGS,
        MAX(c.RATINGS) AS MAX_RATINGS
     FROM
        C_PROFILE pr
     JOIN C_PROFILE_CONTENT pc ON pr.PROFILE_ID = pc.PROFILE_ID
     JOIN C_CONTENT c ON pc.CONTENT_ID = c.CONTENT_ID
     GROUP BY pr.USER_ID) u_stats
WHERE
    p.USER_ID = u_stats.USER_ID;
```

Profile ID	User Name	User ID	Number of Profiles	Number of Contents	Average Rating	Min Ratings	Max Ratings
7	Sachin S	1007	1	2	8.5	8.3	8.7
4	Saumay S	1004	1	1	8.2	8.2	8.2
8	Rohit R	1008	1	2	8.5	8.2	8.8
6	Dhoni M	1006	1	3	8.6	8.3	8.8
1	Rudra Patel	1001	1	2	8.45	8.2	8.7
2	Purva P	1002	1	3	9	8.7	9.5
5	Pranav P	1005	1	3	8.8	8.2	9.5
3	Maanvi R	1003	1	4	8.8	8.3	9.5

Download CSV

8 rows selected.

Business purpose

The purpose of this query is to list a profile's ID, name, and their respective user's total number of profiles, total content count, average content rating, and minimum, and maximum content rating. The query's business purpose is to analyze user engagement by profiling content consumption patterns and inform strategic decisions regarding content management, user experience, and personalized marketing efforts.

Explanation of the Query

In this query:

- We're selecting from the C_PROFILE table.
- An in-line view (subquery) is created to select the count of profiles per user, the total count of content watched per user, and the average, minimum, and maximum ratings of that content. This is achieved by joining the C_PROFILE, C_PROFILE_CONTENT, and C_CONTENT tables and grouping by USER_ID.

- We use a join in the FROM clause (the old comma-separated style, which is equivalent to an inner join) between the C_PROFILE table and the in-line view u_stats based on the USER_ID field.
- We combine the first and last names to create a full name for the profile.
- We truncate the average ratings to 2 decimal places.

LEARNING OUTCOME FROM THE GROUP PROJECT

As a Team

As a team, we learned to navigate the complexities of designing and implementing a comprehensive database system tailored to combat unauthorized access and password sharing. Our collective efforts improved our understanding of advanced data analytics, SQL coding, and security measures. We developed critical teamwork skills, such as effective communication, problem-solving, and project management, which were essential in adapting to challenges and ensuring the project's success. This experience has provided us with valuable insights into the practical application of database technologies in real-world scenarios, preparing us for future collaborative tech projects.

As an Individual Member

Purva Patel

In this group project, I learned the intricacies of database management and the critical role it plays in addressing real-world issues like unauthorized access and password sharing. Collaborating closely with my team allowed me to deepen my understanding of SQL, data modeling, and security protocols. This experience not only enhanced my technical skills but also improved my problem-solving capabilities and team collaboration, equipping me with the tools to effectively tackle complex data security challenges.

Rudra Patel

During the group project, I learned to create relational databases, construct both conceptual and logical data models, and apply normalization to ensure data integrity and efficiency. I also developed skills in transforming complex business requirements into efficient, reliable databases aligned with business objectives, and I gained a deeper understanding of the goals of data and information management. Personally, the project significantly boosted my self-confidence, particularly through the positive feedback and support from my team members. Their encouragement not only enhanced my performance but also highlighted the value of working in a supportive team environment, where each member's contribution is valued and celebrated, fostering innovative ideas and mutual growth.

Saumay Killa

Through this group project, I learned the best practices involved in designing and maintaining a database. This project taught me the difference between logical and relational models and the importance of having them. Participating in a team-based project also provided me with an understanding of the complexities in a collaborative environment. I encountered various challenges and learned effective strategies to address them. Through navigating differing opinions,

conflicting priorities and communication barriers, I gained the ability to resolve conflicts constructively and maintain team unity.

Pranav Bokar

As a member in a Database Management System (DBMS) project, my learning objectives involve understanding the fundamentals of database design, gaining hands-on experience with putting database schemas into practice, being proficient with SQL queries for data retrieval, manipulation, and management, being able to effectively collaborate in a team setting to solve problems, and being flexible enough to adjust to new technologies and approaches in the database management space. By means of practical experience and cooperative problem-solving, my objective is to cultivate a comprehensive skill set that will allow me to make valuable contributions to upcoming DBMS initiatives and grow in database management positions.

Maanvi Reddy

During the group project on database design and management, I learned how to create and manage important databases for the real business world. Throughout the project, I learned how to effectively design a database, by carefully planning the schema and understanding the importance of structured query language (SQL) for manipulating and retrieving data. I learned how to use different functions to retrieve data and also got the chance to share my knowledge with my peers through discussion forums. I also learned how to create and manage databases for Netflix which had issues related to unauthorized access and data security and how comments are necessary while designing a database. Additionally, I gained experience in coordinating tasks and responsibilities within our team. Managing time efficiently was essential, especially when integrating our individual parts into the final database system, ensuring our project's timely completion and functional integrity. This experience has significantly enhanced my technical capabilities and teamwork skills.

Fengxia Yan

Having joined this team, I'm thrilled to have contributed to our shared learning and success in database management. Drawing from the team's outcomes, I've enhanced my understanding of database design, SQL, and security protocols. Collaborating with skilled teammates not only deepened my technical expertise but also refined my teamwork and problem-solving skills. Together, we navigated through intricate challenges, leveraging our collective knowledge and experience to drive innovation and achieve remarkable outcomes. Through our collaborative efforts, we developed comprehensive database solutions tailored to address real-world issues, such as unauthorized access and password sharing. This experience has not only expanded my technical proficiency but also enriched my professional growth, equipping me with invaluable skills and insights for future endeavors in database management. As we reflect on our journey, I am proud of the contributions we've made to the field, and I am grateful for the opportunity to have been a part of such a dynamic and supportive team. Looking ahead, I am excited to continue leveraging our collective strengths to tackle new challenges and drive further innovation in database management.

REFERENCES

Demers, B. (2023) *Survey: Netflix Password Sharing Fees Could Hit Huge Slice of US Market*. 31 May 2023. Kiplinger.com. https://www.kiplinger.com/personal-finance/spending/survey-netflix-password-sharing-fees-could-hit-huge-slice-of-us-market#.