We will discuss the techniques used by the security and authorization subsystem for protecting the DB against persons who are not authorized to access either certain parts of a DB or the whole DB.

It is now customary to refer to two types of DB security

– Discretionary security mechanism
– Mandatory access control

Discuss also the security problem in statistical DBs
Types of security

- Database security is a very broad area that address many issues.
  - Policy issues at the governmental, system-related issues, etc.

1st problem: In a multiuser DBS, the DBMS must provide techniques to enable certain users or group of users to access selected portions of a DB without gaining access to the rest of the DB.

- a DBMS typically includes a **DB security and authorization subsystem** responsible to ensure the security portion of DB.

- Two types of DB security mechanisms.
  - Discretionary security mechanisms
    - Used to grant privileges to users
      - Include capabilities to access specific data files, or records, etc. in a specified mode —read, insert, etc.—
  - Mandatory security mechanisms
    - Used to enforce multilevel security.
      - Classifying the data and users into various security classes —levels—.
        - A typical security policy of an organization is to allow certain classification level to see only the data items classified at the user’s own (or lower) classification level.
Types of security (cont’d)

2nd problem: -problem common to all computer systems- prevent unauthorized persons from accessing the system itself as a whole–either to obtain information or to make changes-

The DBMS offer the access method function to restrict access to the DBS as a whole.

– Access method handled by creating user accounts and passwords to control the log-in process by the DB.

3rd problem: related to statistical DB security.

– Ensure that confidential information can’t be retrieved even by deduction.

4th problem: data encryption used to protect sensitive data being transmitted via a network.

– Encode data using some coding algorithm.
– Authorized users can decipher the data using the decoding (or crypting) algorithm (or keys).
– Unauthorized users who access the encoded data will have difficulty deciphering it.
DBA Security and the DBA

- The DBA is the central authority for managing a DBS. His responsibilities include
  – Granting privileges to users who need to use the system
  – Classifying users and data in accordance with policy of the organization.

- DBA has a DBA account in the DBMS –called system or super account- which provides privileged commands
  - Account creation –Control access to DBS-
    – Creates a new account and password for a user or group of users to enable them to access the DBMS.
  - Privilege granting –Control discretionary DB authorization-
    – Grant certain privileges to certain accounts.
  - Privilege revocation -Control discretionary DB authorization-
    – Revoke, or cancel, certain privileges previously given to certain users.
  - Security level assignment –Control mandatory authorization-
    – Assigning users accounts to the appropriate security classification level
Access Protection, User Accounts, and Database Audits

Whenever a person or group of persons needs to access the DBS

• First, the user or group of users apply for a user account.
• Then the DBA will create a new account number and password.
• After, the user must log-in to the DBMS by entering the account number and password whenever DB access is needed.
  – Application programs are also be considered as users and can be required to supply passwords
• The DBMS checks that the account number and password are valid.
  – If they are, the user allowed to use the DBMS and to access the DB.

Keep track of all operations on the DB applied by a certain
  – Expand the system log entries to include the account number and the on-line terminal ID of the user for each operation applied by that user.

Perform a Database audit if any tampering of the DB is suspected.
  – Review the log to examine all accesses and operations to the DB during a certain period of time
Discretionary Access Control

• The typical method of enforcing discretionary access control in DBS is based on granting and revoking of privileges.
• Many current relational DBMS include additional statements in the query language to allow the DBA and selected users to grant and revoke privileges.
• The DBMS must provide selective access to each relation in the DB based on specific accounts.
  • Operations on relations must also be controlled.
  • Thus having an account does not necessarily entitle the account holder to all functionality provided by the DBMS
– In SQL2, the word authorization identifier, user account or user can be used interchangeably.
Discretionary Access Control (cont’d)

• There are two levels for assigning privileges to use the DBS

  – Account level

  • The DBA specifies the particular privileges that each account holds independently of the relations in the database. Such capabilities provided to the account itself can be:

    – CREATE SCHEMA or CREATE TABLE privilege –to create a schema or base relation-, CREATE VIEW privilege, ALTER privilege –to apply changes such as adding attributes to relations-, DROP privilege -drop tables or views-, MODIFY privilege –insert, delete or update tuples-, SELECT privilege –to retrieve information from the DB using SELECT query-.

      ❑ If an account doesn’t have the CREATE TABLE privilege, no relation can be created from that account.

  • Account privileges are not defined as part of SQ2 and left to the DBMS implementation to define.
Discretionary Access Control (cont’d)

– Relation level

  • We can control the privilege to access each individual relation or view in the database
  • Privileges at the relation level specify for each user the individual relations or views on which each type of command can be applied.
  • Some privileges refer to individual columns (attributes)
  • Relation level privileges are part of SQL2.
  • The granting and revoking of privileges follows the **access matrix model**
    – Rows of matrix M represent subjects –users, accounts, programs- and columns represent objects –relations, records, columns, views, operations-.
    – Each possible M(i,j) represents the type of privileges (read, write, update) that subject i holds on object j.
Discretionary Access Control (cont’d)

– Relation level (cont’d)

• Each relation or view in the DB is assigned an owner account which is the account that was used when the relation was created in the first place.
• The owner of a relation or view is given all privileges on that object
• The owner account can pass privileges on any of the owned relations or views to other users by granting privileges to their accounts.
• The following types of privileges can be granted on each individual relation or view R

  – SELECT (retrieval or read) privilege on R: gives the account retrieval privilege. In SQL this gives the account the privilege to use the SELECT statement to retrieve tuples from R.
  – MODIFY privilege on R: gives the account the capability to modify tuples of R. In SQL this privilege is further divided into UPDATE, DELETE, and INSERT privileges to apply the corresponding SQL command to R. In addition, both INSERT and UPDATE privileges can specify that only certain attributes of R can be updated by the account.
  – REFERENCE privilege on R: gives the account the capability to reference relation R when specifying integrity constraints. This privilege can also be restricted to specific attributes of R.
Discretionary Access Control (cont’d)

• The mechanism of views is an important discretionary authorization mechanism.
  – If the owner A of a relation R wants another account B to be able to retrieve only some fields of R, then A can create a view V of R that includes only those attributes and then grant SELECT on V to B

• To create a view, the account must have SELECT privilege on all relations involved in the view definition.
Discretionary Access Control (cont’d)

• In some cases it is desirable to grant some privileges to a user temporarily.
  – Hence a mechanism for revoking privileges is needed.
  – In SQL a REVOKE command is included to cancel privileges

• If a privilege is granted with GRANT option to an account
  – This account can also grant that privilege on the relation to other accounts.
  – Suppose that B is given the GRANT OPTION by A and that B then grants
    the privilege on R to a third account C, also with GRANT OPTION. In this
    way, privileges on R can propagate to other accounts without the knowledge
    of the owner of R. If the owner account A now revokes the privilege granted
    to B, all the privileges that B propagated based on that privilege should
    automatically be revoked by the system.
  – It is possible for a user to receive a certain privilege from two or more
    sources. In this case the user will continue to have the privilege until all the
    sources revoke the user’s privilege
    • A4 may receive a certain UPDATE R privilege from both A2 and A3. If A2 revokes the
      privilege from A4, A4 still continue to have the privilege given to it by A3
Discretionary Access Control (cont’d)

- CREATE SCHEMA Example AUTHORIZATIONA1;
- User account A1 can create tables under the schema called Example.
- Suppose that A1 creates two base relations EMPLOYEE and DEPARTMENT as shown below:
  - EMPLOYEE (NAME, SSN, BDATE, ADDRESS, SEX, SALARY, DNO)
  - DEPARTMENT(DNUMBER, DNAME, MGRSSN)
- A1 is the owner of these two relations and hence has all the relation privileges on each of them
- GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
- A1 grants account A2 the privilege to insert and delete tuples in both of these relations (A1 does not want A2 to be able to propagate these privileges to additional accounts).
Discretionary Access Control (cont’d)

- GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION;
- A1 wants to allow account A3 to retrieve information from either of the two tables and also to be able to propagate the SELECT privilege to other accounts.
- A3 can grant the SELECT privilege on the EMPLOYEE relation to A4
  GRANT SELECT ON EMPLOYEE TO A4;
- Now suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 then can issue this command
  REVOKE SELECT ON EMPLOYEE FROM A3;
- The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4.
Discretionary Access Control (cont’d)

– A1 wants to give back A3 a limited capability to SELECT from the EMPLOYEE relation and wants to allow A3 to be able to propagate the privilege. The limitation is to retrieve only the NAME, BDATE, and ADDRESS attributes and only for the tuples with DNO=5. A1 then can create the following view:

CREATE VIEW A3EMPLOYEE AS
  SELECT NAME, BDATE, ADDRESS FROM EMPLOYEE WHERE DNO=5;
GRANT SELECT ON A3EMPLOYEE TO A3 WITH GRANT OPTION;

– Suppose A1 wants to allow A4 to update only the SALARY attribute of EMPLOYEE;

GRANT UPDATE ON EMPLOYEE (SALARY) TO A4;

– The UPDATE or INSERT privilege can specify particular attributes that may be updated or inserted in a relation

– Other privileges (SELECT, DELETE) are not attribute-specific.
Mandatory Access Control

- Most of the commercial DBMS currently provide mechanisms only for discretionary access control.
- However in many applications, an additional security policy is needed that classifies data and users based on security classes –known as mandatory access control-
- Typically the techniques can be combined.
- Typical security classes:
  - Top Secret (TS).
  - Secret (S).
  - Confidential (C).
  - Unclassified (U).
  - TS is the highest level and U the lowest level
    - TS > S > C > U.
Mandatory Access Control (cont’d)

- One of the commonly used model for multilevel security is known as Bell-LaPadula model.
  - Classifies each subject (user, account, program) and object (relation, tuple, column, view, operation) into one of the security classifications TS, S, C, or U.
    - Refer to the clearance (classification) of a subject $S$ as $\text{class}(S)$ and to the classification of an object $O$ as $\text{class}(O)$.
  - Two restrictions are enforced on the Subject/Object classifications:
    1. A subject $S$ is not allowed read access to an object $O$ unless $\text{class}(S) \Rightarrow \text{class}(O)$.
      - Known as simple security property
    2. A subject $S$ is not allowed to write an object $O$ unless $\text{class}(S) \leq \text{class}(O)$.
      - Known as the *-property (star rule).
Mandatory Access Control (cont’d)

• Simple security rule is intuitive
  – Enforce the obvious rule that no subject can read an object whose security classification is higher than the subject’s security clearance.

• *-property is less intuitive
  – Prohibits a subject from writing an object at a lower security classification than the subject’s security clearance. Violation of this rule would allow information to flow from higher to lower classifications.
    • A user with TS clearance, for example, may make a copy of an object with classification TS and write it back as a new object with classification U, thus making it visible throughout the system.
Mandatory Access Control (cont’d)

• Multilevel security and the relational database model
  – Consider attributes values and tuples as data object.
  – Each attribute A is associated with a classification attribute C in the schema.
  – Each attribute value in a tuple is associated with a corresponding security classification.
  – In some models, a tuple classification attribute TC is added to the relation attributes to provide a classification for each tuple as a whole.
  – A multilevel relation schema R with n attributes would be represented
    \[ R(A_1, C_1, A_2, C_2, \ldots, A_n, C_n, TC) \]
    where \( C_i \) represents the classification attribute associated with attribute \( A_i \)
  – The value of TC attribute for each tuple t is the highest of all attribute classification values within t.
  – Value of TC attribute for a tuple t provides a general classification for the tuple itselfs, whereas each \( C_i \) provide a finer security classification for each attribute value within the tuple.
  – The apparent key of a multilevel relation is the set of attributes that would have formed the primary key in a regular (single-level) relation.
Mandatory Access Control (cont’d)

• Multilevel security and relational model (cont’d)
  – A multilevel relation will appear to contain different data to subjects (users) with different clearance levels.
  – In some cases, it is possible to store a single tuple in the relation at a higher classification level and produce the corresponding tuples at a lower level classification using a process known as filtering.
  – In other cases, it is necessary to store one or more tuples at different classification levels with the same value for the apparent key: concept of polyinstantiation where several tuples can have the same apparent key value but have different attribute values for users at different classification levels.
  – The entity integrity rule for the multilevel relations states that all attributes are member of the apparent key must not be null and must have the same security classification within each individual tuple.
    • In addition, all other attribute values in the tuple must have a security classification greater than or equal to that of the apparent key.
  – Null integrity and interinstance integrity ensure that if a tuple value at some security level can be filtered (derived) from a higher-classified tuple, then it is sufficient to store the higher-classified tuple in the multilevel relation.
Mandatory Access Control (cont’d)

• Assume that the Name attribute is the apparent key and consider the query SELECT * FROM EMPLOYEE
  – A user with security clearance would see the same relation shown in (a).
  – A user with clearance C would not be allowed to see values Salary of Brown and JobPerformance of Smith appearing as null – higher classification- like shown in (b)
  – for a user with clearance U, the filtering allows only the name attribute of Smith to appear with all other attributes appearing as null (c).
  – the filtering introduces null values for attribute values whose security classification is higher than the user’s security clearance
Mandatory Access Control (cont’d)

(a) 
<table>
<thead>
<tr>
<th>Name</th>
<th>Salary</th>
<th>JobPerformance</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>U 40000</td>
<td>C Fair</td>
<td>S</td>
</tr>
<tr>
<td>Brown</td>
<td>C 20000</td>
<td>S Good</td>
<td>C</td>
</tr>
</tbody>
</table>

(b) 
<table>
<thead>
<tr>
<th>Name</th>
<th>Salary</th>
<th>JobPerformance</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>U 40000</td>
<td>C Null</td>
<td>C</td>
</tr>
<tr>
<td>Brown</td>
<td>C null</td>
<td>C Good</td>
<td>C</td>
</tr>
</tbody>
</table>

(c) 
<table>
<thead>
<tr>
<th>Name</th>
<th>Salary</th>
<th>JobPerformance</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>U null</td>
<td>U Null</td>
<td>U</td>
</tr>
</tbody>
</table>

(d) 
<table>
<thead>
<tr>
<th>Name</th>
<th>Salary</th>
<th>JobPerformance</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>U 40000</td>
<td>C Fair</td>
<td>S</td>
</tr>
<tr>
<td>Smith</td>
<td>U 40000</td>
<td>C Excellent</td>
<td>C</td>
</tr>
<tr>
<td>Brown</td>
<td>C 20000</td>
<td>S Good</td>
<td>C</td>
</tr>
</tbody>
</table>

• Suppose that a user with security clearance C tries to update the value of JobPerformance of Smith to ‘Excellent’

  UPDATE EMPLOYEE SET JobPerformance = ‘Excellent’
  where Name= ‘Smith’;
  - The system should not reject the query since the view provided to users with clearance C permits such update (c ).
  ☑ Otherwise the user could infer that some nonnull value exists for the JobPerformance attribute of Smith. The inferring should not be allowed.
  - However, the user should not be allowed to overwrite the existing value of JobPerformance at the higher classification level.
  - Create a plyinstantiation for the Smith tuple at the lower classification level C (d). This is necessary since the new tuple cannot be filtered from the existing tuple at classification S.