Part 12: Datenschutz: Zugriffsrechte in SQL

Literatur:

- Elmasri/Navathe:Fundamentals of Database Systems, 3rd Edition, 1999. Chap. 22, "Database Security and Authorization"
- Silberschatz/Korth/Sudarshan: Database System Concepts, 3rd Edition, McGraw-Hill, 1999. Section 19.1, "Security and Integrity"
- Kemper/Eickler: Datenbanksysteme (in German), Ch. 12, Oldenbourg, 1997.
- Lipeck: Skript zur Vorlesung Datenbanksysteme (in German), Univ. Hannover, 1996.
- Date/Darwen: A Guide to the SQL Standard, Fourth Edition, Addison-Wesley, 1997.
- van der Lans: SQL, Der ISO-Standard (in German, there is an English version), Hanser, 1990.
- Oracle8 SQL Reference, Oracle Corporation, 1997, Part No. A58225-01.
- Oracle8 Concepts, Release 8.0, Oracle Corporation, 1997, Part No. A58227-01.
- Don Chamberlin: A Complete Guide to DB2 Universal Database. Morgan Kaufmann, 1998.
- Microsoft SQL Server Books Online: Accessing and Changing Data, Administering SQL Server.
- K. Nagel: Informationsbroschüre zum Bundesdatenschutzgesetz, ("Information about the German Federal Data Privacy Law", in German), 10. Auflage, Oldenbourg, 2001.

Lernziele

Nach diesem Kapitel sollten Sie Folgendes können:

- Einige Vorgehensweisen von Hackern erklären.
- Sicherheitsrelevante DBMS Funktionen erklären.
- Die SQL Befehle `GRANT` und `REVOKE` benutzen.
- Einige Grundzüge des deutschen Datenschutzrechtes beim Aufbau von Datenbanken berücksichtigen.

  Oder zumindest erkennen, wann Sie weitere Informationen einholen müssen.

Inhalt

1. Anforderungen, Allgemeines
2. Benutzer-Authentifikation
3. Datenschutzrecht
4. Die SQL-Befehle GRANT und REVOKE
5. Sichten

DB-Sicherheit: Motivation (1)

- Information: wichtiger Aktivposten von Firmen. Darf nicht in die Hände der Konkurrenz gelangen.

  Z.B. Kundendaten, Einkaufspreise, Produkt-Zusammensetzungen.

- Es gibt Gesetze zum Schutz vertraulicher Daten.

  Wenn man z.B. zuläßt, daß ein Hacker Zugriff auf Kreditkartennummern bekommt und diese Daten dann mißbraucht, können hohe Schadenersatzforderungen entstehen. Außerdem: schlechte Reputation.

- Die Vertraulichkeit bestimmter Daten muß auch innerhalb der Firma geschützt werden (z.B. Gehälter).

  Man muß auch verhindern, daß Angestellte einfach alle Daten mitnehmen können, wenn sie die Firma verlassen.

DB-Sicherheit: Motivation (2)

- Falls es einem Hacker gelingen sollte, alle Daten zu löschen, oder vollständig durcheinander zu bringen, entsteht ein gewaltiger Schaden für die Firma.

  Man sollte mindestens noch Sicherungskopien haben, auch offline, und an einem anderen Ort untergebracht (wegen einem möglichen Brand im Rechnerraum). Aber die Wiederherstellung dauert, und die letzten Änderungen sind schwer zu rekonstruieren.

- Unautorisierte Änderungen / Verfälschungen der Daten müssen verhindert werden, z.B. sollten Angestellte nicht ihr eigenes Gehalt ändern können.

  Bestimmte Geschäftsregeln — wer darf was tun — sollten durch das Informationssystem automatisch sichergestellt werden.

Sicherheits-Anforderungen (1)

Es sollte möglich sein . . .

- sicher zu stellen, daß nur die tatsächlich legitimierten Benutzer Zugriff auf die Datenbank haben.

  → Benutzer Identifikation/Authentifikation

- festzulegen, welcher Nutzer welche Operationen auf welchen DB-Objekten durchführen darf.

  → Benutzer-Autorisierung

- die Aktionen der Benutzer zu protokollieren.

  → Auditing

Sicherheits-Anforderungen (2)

Es sollte auch möglich sein . . .

- die Benutzung von Resourcen (Platten-Platz, CPU-Zeit) für jeden Nutzer zu begrenzen (Quotas).

  Sonst kann ein einzelner Nutzer die ganze Datenbank zum Stillstand bringen.

- Gruppen von Benutzern mit den gleichen Rechten zu verwalten.

  Umgekehrt kann ein Nutzer auch unterschiedliche Rollen haben.

- mehrere Administratoren mit unterschiedlichen Befugnissen zu haben (nicht nur einen einzelnen Nutzer “root”, der alles darf).

Sicherheits-Anforderungen (3)

Es sollte auch möglich sein . . .

- die Vertraulichkeit und Integrität der Daten auch in einer Netzwerk-Umgebung zu garantieren.

  Oft geschieht die Client-Server Kommunikation unverschlüsselt.

- sicher zu stellen, daß niemand direkten Zugriff auf die Daten hat (und damit die Zugriffskontrolle der Datenbank umgeht).

- darauf zu vertrauen, daß nicht durch Programmierfehler im DBMS Sicherheitslücken bestehen (oder sie wenigstens schnell korrigiert werden).

Zugriffsrechte (1)

- Oft haben verschiedene Nutzer einer Datenbank auch verschiedene Rechte.
- Kommandos bestehen aus
  - ◇ "Subject" (der Benutzer, der es ausführt),
  - ◇ "Verb" (die Operation, z.B. "`INSERT`), und
  - ◇ "Objekt" (typischerweise eine Tabelle).
- In SQL kann man mit dem `GRANT`-Kommando (s.u.) definieren, welche dieser Tripel erlaubt sind.

  Das DBMS speichert also eine Menge von Tripeln (u, o, d), aus Benutzer u, Operation o (im wesentlichen `SELECT`, `INSERT`, `UPDATE`, `DELETE`), Objekt/Tabelle o. Eigentlich Quadrupel: Wer hat das Recht erteilt?

Zugriffsrechte (2)

- Das Subjekt-Verb-Objekt Modell für die Zugriffsrechte hat aber Einschränkungen:
  - ◇ Kommandos wie "`CREATE TABLE`" beziehen sich nicht auf existierende Objekte, aber man muß ihre Benutzung einschränken können.

    Prinzip: Jeder sollte nur das tun können, was er tun muß.
  - ◇ Für DB-Administratoren sollen die normalen Zugriffsbeschränkungen nicht gelten. Es soll aber auch nicht jeder Administrator alles können.

    Große Firmen haben mehrere Administratoren.

Zugriffsrechte (3)

- Während das Subjekt-Verb-Objekt Modell sehr portabel ist (schon im SQL-86 Standard enthalten), sind die Erweiterungen zur Lösung der obigen Probleme meist vorhanden, aber sehr systemabhängig:
  - ◇ Oracle hat neben Objektrechten nach dem obigen Modell noch eine große Anzahl von Systemrechten (wer darf welches Kommando/Verb benutzen, ggf. auch für beliebige Objekte).
  - ◇ Andere Systeme unterscheiden oft nur wenige Benutzerklassen, die verschieden privilegiert sind.

Zugriffsrechte (4)

- Sichten und serverseitige Prozeduren ("stored procedures") erlauben es, Objekte zu verkapseln:
  - ◇ Z.B. ist es möglich, daß ein Benutzer für eine Tabelle selbst kein `SELECT`-Recht hat, aber über eine Sicht bestimmte Zeilen/Spalten oder aggregierte Daten sehen darf.
  - ◇ Es ist auch möglich, daß ein Benutzer kein direktes `INSERT`-Recht für eine Tabelle hat, aber eine Prozedur benutzen kann, die Daten in diese Tabelle nach zusätzlichen Prüfungen einfügt.

Sicherheitsmodelle

- Es gibt zwei grundlegend verschiedene Sicherheitsmodelle. Normale Datenbanksysteme implementieren nur das erste:
  - ◇ "Discretionary Access Control" erlaubt den Administratoren, die Zugriffsrechte nach Belieben zu vergeben.
  - ◇ "Mandatory Access Control" basiert auf einer Klassifizierung von Benutzern und Daten.

    Z.B. "confidential", "secret", "top secret". Benutzer können nur Daten auf der eigenen oder einer niedrigeren Sicherheitsebene lesen, und nur Daten auf der eigenen oder einer höheren Sicherheitsebene schreiben.

Auditing

- In manchen Systemen kann man die Aktionen der Benutzer mitprotokollieren lassen.

  Das gibt typischerweise Ärger mit dem Betriebsrat, eventuell auch mit dem Datenschutz. Man beachte, daß nicht nur erfolgreich ausgeführte Kommandos protokolliert werden müssen, sondern auch Kommandos, die z.B. wegen nicht ausreichender Zugriffsrechte abgelehnt wurden.
- So kann man wenigstens hinterher herausfinden, wer für ein Problem verantwortlich ist.
- Eventuell fallen bei einer Kontrolle auch merkwürdige Muster auf, bevor wirklich ein Schaden eintritt.

  Man muß sehr selektiv mitprotokollieren, wenn man im Protokoll wirklich noch etwas Interessantes finden will.

Datensicherheit

- Niemand (der nicht ohnehin DBA Rechte hat) sollte direkten Zugriff auf die Daten haben (und damit die Zugriffskontrolle der Datenbank umgehen).

  Aus Performance-Gründen werden die Daten normalerweise unverschlüsselt in Betriebssystem-Dateien gespeichert. Wer auf diese Dateien Zugriff hat, kann auch ohne DB-Account die Daten darin lesen.
- Backup Bänder müssen weggeschlossen werden.
- In Deutschland wurde ein PC gestohlen, der ein AIDS-Register enthielt. In den USA sind mehrere Notebooks der CIA mit geheimen Unterlagen verschwunden.

Inhalt

Benutzer-Authentifikation (1)

- Üblich ist, daß sich Benutzer mit Passworten gegenüber dem System ausweisen.
- Zum Teil führen die DBMS selbst eine Benutzer-Identifikation durch, zum Teil verlassen sie sich auf das Betriebssystem.

  Beide Möglichkeiten haben Vor- und Nachteile: Es ist bequem, nicht mehrfach Passworte eingeben zu müssen, und es ist besser, wenn Anwendungsprogramme keine Passworte enthalten. Aber in einer Client-Server-Umgebung kann man nicht so sicher sein, daß der Benutzer auf dem Client tatsächlich die Person ist, die er/sie behauptet, zu sein (und daß der Computer wirklich der richtige Computer ist). Manche Client-Betriebssysteme haben ein schwächeres Sicherheitssystem als der Server (bei manchen PCs kann jeder eine Boot-CD einlegen).

Benutzer-Authentifikation (2)

- Es ist gefährlich, wenn des gleiche Passwort immer wieder genutzt wird.

  Wenn ein Hacker das Passwort irgendwie beobachten kann, kann er es auch nutzen. Banken senden ihren Kunden typischerweise Listen mit Einmal-Passworten (TANs).
- Einige Administratoren zwingen die Nutzer, ihre Passworte in regelmäßigen Abständen zu ändern.

  Z.B. jeden Monat. Dies kann aber zu schwächeren Passworten führen, (und zu häufigeren Problemen mit vergessenen Passworten), als wenn der Benutzer ein neues Passwort nur wählt, wenn er dazu bereit ist. Wenn ein System den Benutzer zwingt, sein Passwort zu ändern, prüft es üblicherweise auch, daß neues und altes Passwort sich deutlich unterscheiden, und daß der Benutzer nicht zu schnell zurück wechselt.

Benutzer-Authentifikation (3)

- Trojaner sind Programme, die neben der Hauptfunktion, für die sie eingesetzt werden, im Verborgenen noch andere (schlechte) Dinge tun.
- Speichern Sie nie ein Passwort für einen anderen Computer auf der Festplatte Ihres Computers. Das gilt auch für Cookies, die Passworte ersetzen.

  Wenn Sie sich auf diese Art bei dem anderen Rechner einloggen können, ohne ein Passwort einzugeben, kann das auch jemand, der sich (z.B. mit Trojaner) die entsprechenden Daten von Ihrem System besorgt. Wenn das Passwort in einer Datei steht, ist es besonders einfach, und betrifft auch Passworte, die Sie nicht nutzen, wenn der Trojaner aktiv ist. Je nach Betriebssystem könnte er aber auch Tastendrücke mitloggen oder den Hauptspeicher anschauen.

Benutzer-Authentifikation (4)

- Passworte sollten nicht leicht zu raten sein.

  Der Name Ihrer Freundin/Ihres Freundes, Ihrer Lieblings-Popgruppe (Autor, Schauspieler, Film, Urlaubsland, etc.), Ihre Telefonnummer, Adresse, Geburtsdatum wären besonders leicht. Falls es einem Hacker gelingt, an eine verschlüsselte Version Ihres Passwortes zu kommen, kann er mit einer schnellen Verschlüsselungsroutine viele Worte probieren (selbst wenn eine direkte Entschlüsselung nicht möglich ist). Ein Hacker würde in so einem Fall den Duden durchprobieren, alle Namen von Personen, Popgruppen, relativ lange Folgen von Ziffern, zumindest alle kurzen Folgen von Kleinbuchstaben, etc. Außerdem alles auch rückwärts und auf verschiedene andere Arten leicht verändert. Deswegen wird empfohlen, daß ein gutes Passwort nicht zu kurz sein sollte, und Buchstaben, Ziffern, und Sonderzeichen enthalten sollte. Um eine möglichst zufällige Buchstabenfolge zu bekommen, kann man sich einen Satz denken und davon die Anfangsbuchstaben nehmen.

Benutzer-Authentifikation (5)

- Wenn ein System mehrere Anmeldeversuche mit falschem Passwort entdeckt, sollte es einen Alarm auslösen und eventuell den Account sperren.

  Außerdem gibt es oft eine künstliche Verzögerung, bevor das Programm dem Benutzer mitteilt, daß die Anmeldedaten falsch waren. So kann ein Hackerprogramm nicht viele Worte in kurzer Zeit probieren. Das Sperren des Accounts bestraft aber den korrekten Nutzer.

- Viele DBMS haben Default-Passworte für den Administrator. Diese müssen sofort geändert werden.

  Z.B. hat in Oracle der automatisch angelegte Administrator-Account `SYSTEM` das Passwort `MANAGER`, und das Passwort für `SYS` (kann alles) ist `CHANGE_ON_INSTALL`. Jeder Hacker weiß das. In SQL Server hat der mächtigste Account `sa` direkt nach der Installation kein Passwort.

Benutzer-Authentifikation (6)

- Man sollte Gast-Accounts löschen oder sperren.

  Oracle hat einen Account `SCOTT` mit Passwort `TIGER`. SQL Server hat einen Account `guest`, für Windows-Nutzer, die der Datenbank unbekannt sind. Prüfen Sie die Accounts im System in regelmäßigen Abständen und sperren Sie alle, die nicht wirklich benötigt werden.

- Das die meisten DBMS Client-Server Systeme sind, ist der DB-Server automatisch am Netz, sobald Ihr Rechner mit dem Internet verbunden ist.

  In dieser Hinsicht verhält sich der DB-Server ähnlich wie ein Web-Server. Selbst wenn der Hacker sich nicht beim Betriebssystem anmelden kann, kann er sich vielleicht bei der Datenbank anmelden. Oracle wartet normalerweise auf Port 1521 auf Verbindungen, die Default-SID ist "`ORCL`".

Benutzer-Authentifikation (7)

- Wird ein Passwort unverschlüsselt über das Internet verschickt, können es Hacker eventuell lesen.
  - ◇ Beim klassischen Ethernet kann man alle Pakete mitlesen, die über das Netz geschickt werden.

    Mit modernen Switches ist das nicht mehr möglich, aber erst, nachdem der Switch die Adressen der angeschlossenen Rechner gelernt hat. Man kann sich also nicht darauf verlassen, daß andere Rechner, die an das lokale Netz angeschlossen sind, die Datenpakete nicht auch erhalten.
  - ◇ Die Route, die Datenpakete über das globale Internet nehmen, ist kaum vorhersehbar.

    Ein Gateway wird vielleicht von einem Hacker betrieben, oder ist schon von einem Hacker "geknackt".

Benutzer-Authentifikation (8)

- Wenn Sie ein Passwort eingeben, achten Sie darauf, daß Sie auch mit dem richtigen Programm bzw. dem richtigen Computer verbunden sind.

  Früher gab es Programme, die wie ein UNIX Login Prompt aussahen, aber das Passwort an einen Hacker schickten.

  Der Suchpfad für Kommandos darf nur vertrauenswürdige Verzeichnisse enthalten (z.B. nicht "."). Sonst bekommt man vielleicht ein ganz anderes Programm, wenn man z.B. "sqlplus" aufruft.

  Es gibt viele "Phishing" ("password fishing") EMails, die z.B. behaupten, von der eigenen Bank zu kommen, und zur Eingabe von PIN und TAN auffordern. Tatsächlich kommt man mit der URL aber auf eine Hacker-Webseite, die wie die echte Webseite der Bank aussieht.

Benutzer-Authentifikation (9)

- Man sollte möglichst nicht das gleiche Passwort für verschiedene Systeme (auch Webshops) nutzen.

  Angestellte des Webshops können Ihr Passwort möglicherweise im Klartext lesen.

- Sagen Sie Ihr Passwort (auch PIN etc.) nie offiziell klingenden Unbekannten am Telefon!

  Z.B. dem technischen Support des Rechenzentrums, das gerade auf LDAP umstellt, und Ihnen die Mühe ersparen möchte, ins Rechenzentrum zu kommen, und dort Ihr Passwort noch einmal einzutippen.

Inhalt

Datenschutzrecht (1)

- In diesem Abschnitt sollen die Vorschriften des Bundesdatenschutzgesetzes erklärt werden.

  Ich bin kein Jurist, und der zur Verfügung stehende Platz ist nicht ausreichend für eine vollständige Erklärung. Alle Angaben sind ohne Gewähr. Wenn es wirklich wichtig ist, informieren Sie sich bitte bei einem Juristen.

- Das Gesetz soll davor schützen, daß man durch Umgang mit seinen personenbezogenen Daten in seinem verfassungsmäßig garantierten Persönlichkeitsrecht beeinträchtigt wird (freie Selbstbestimmung bei der Entfaltung der Persönlichkeit).

Datenschutzrecht (2)

- Es soll vermieden werden, daß “Bürger nicht mehr wissen können, wer was wann bei welcher Gelegenheit über sie weiß.”
- Man soll normalerweise selbst über die Preisgabe und Verwendung seiner personenbezogenen Daten bestimmen können (“Informationelle Selbstbestimmung”).
- Personenbezogene Daten sind “Einzelangaben über persönliche oder sächliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person”.

Datenschutzrecht (3)

- Das Gesetz bezieht sich nicht auf aggregierte oder anonymisierte Daten.
- Ebenfalls ausgenommen sind Daten, die ausschließlich für persönliche oder familiäre Tätigkeiten erhoben/verarbeitet werden.
- Das Gesetz gilt auch für "nicht-automatisierte Dateien".

  Es ist also nicht richtig, daß das Gesetz grundsätzlich erst dann anwendbar ist, wenn die Daten in einem Computer gespeichert werden. Die Daten müssen aber gleichartig aufgebaut sein und nach bestimmten Merkmalen zugänglich sein (Kartei).

Datenschutzrecht (4)

Besonders schützungswerte, sensitive Daten:

- "Besondere Arten personenbezogener Daten" sind:
  - rassische und ethnische Herkunft
  - politische Meinungen
  - religiöse oder philosophische Überzeugungen
  - Gewerkschaftszugehörigkeit
  - Gesundheit
  - Sexualleben
- Für diese Daten gelten teilweise verschärfte Vorschriften (s.u.). Sie müssen in jeder Einverständniserklärung explizit genannt werden.

Datenschutzrecht (5)

Verbot mit Ausnahmen:

- Personenbezogene Daten dürfen nur abgespeichert werden, wenn
  - der Betroffene zugestimmt hat,
  - die Daten aufgrund eines Gesetzes gespeichert werden müssen,
  - die Daten zur Erfüllung eines Vertrages (o.ä.) mit dem Betroffenen gespeichert werden müssen,
  - ... (Fortsetzung siehe nächste Folie)

Datenschutzrecht (6)

Verbot mit Ausnahmen, Forts.:

- Zulässige Gründe für die Speicherung personenbezogender Daten, Forts.:
  - wenn es zur Wahrung berechtigter Interessen der Firma nötig ist, und es keinen Grund zur Annahme gibt, daß schutzwürdige Interessen des Betroffenen überwiegen,
  - die Daten allgemein zugänglich sind oder veröffentlicht werden dürften.

    Sofern keine schutzwürdigen Interessen entgegenstehen.

Datenschutzrecht (7)

Zweckbindung der Daten:

- Daten müssen für einen speziellen Zweck gesammelt werden und dürfen später nur unter bestimmten Voraussetzungen für andere Zwecke verwendet werden:
  - ◇ Wahrung berechtigter Interessen (wie oben)
  - ◇ Daten allgemein zugänglich (wie oben)
  - ◇ Wahrung berechtigter Interessen eines Dritten,

    Mit Einschränkungen, z.B. bei strafbaren Handlungen, oder Angaben des Arbeitgebers auf arbeitsrechtliche Verhältnisse.

Datenschutzrecht (8)

Zweckbindung der Daten, Forts.:

- Gründe für Nutzung von Daten zu anderen Zwecken als bei der ursprünglichen Erhebung, Forts.:
  - ◇ Abwehr von Gefahren für die staatliche Sicherheit, Verfolgung von Straftaten
  - ◇ Werbung/Marktforschung mit Einschränkungen

    Wenn es sich eine Liste von Angehörigen einer Personengruppe handelt, die nur die Zugehörigkeit zu der Gruppe, Berufsbezeichnung, Namen, akademische Titel, Anschrift und Geburtsjahr enthält, und kein Grund zur Annahme besteht, daß der Betroffene ein schutzwürdiges Interesse hat, das dem entgegensteht.
  - ◇ Für Forschungszwecke (unter Bedingungen).

Datenschutzrecht (9)

Benachrichtigungspflicht:

- Personen müssen darüber informiert werden,
  - ◇ daß Daten über sie gespeichert werden,
  - ◇ welche Arten von Daten das sind,
  - ◇ was der Zweck der Datenverarbeitung ist,
  - ◇ wer die Daten speichert (Stelle/Firma)

  (falls sie es nicht ohnehin schon wissen).
- Es gibt aber viele Ausnahmen von der Benachrichtigungspflicht, siehe nächste Folie.

Datenschutzrecht (10)

Ausnahmen von der Benachrichtigungspflicht:

- Daten müssen per Gesetz gespeichert werden
- Daten entstammen einer öffentlichen Quelle, unverhältnismäßiger Aufwand
- Daten müssen geheim gehalten werden (Gesetz oder überwiegendes Interesse eines Dritten)
- Würde Geschäftszweck erheblich gefährden (und kein überwiegendes Interesse des Betroffenen)
- Werbezwecke mit eingeschränkten Daten (s.o.), unverhältnismäßiger Aufwand.

Datenschutzrecht (11)

Datenschutzbeauftragter:

- Firmen oder öffentliche Stellen, die personenbezogende Daten verarbeiten, müssen einen Datenschutzbeauftragten bestellen.

  Bei Firmen gilt das nur, wenn mindestens 10 Personen ständig mit der Verarbeitung dieser Daten beschäftigt sind. Handelt es sich aber um besonders schützenswerte Daten, oder handelt die Firma mit den Daten, ist ein Datenschutzbeauftragter unbedingt nötig.

- Der Datenschutzbeauftragte muß die erforderliche Fachkunde und Zuverlässigkeit besitzen.
- Er ist in Ausübung dieser Tätigkeit weisungsfrei.

Datenschutzrecht (12)

Datenschutzbeauftragter, Forts.:

- Der Datenschutzbeauftragte
  - überwacht die ordnungsgemäße Anwendung der Datenverarbeitungsprogramme,
  - schult die Benutzer in den Vorschriften des Datenschutzes,
  - ist rechtzeitig über Vorhaben zur Verarbeitung personenbezogener Daten zu informieren.
- Es gibt auch externe Berater als Datenschutzbeauftragte.

Datenschutzrecht (13)

Meldepflicht:

- Die Verarbeitung personenbezogener Daten ist den zuständigen Aufsichtsbehörden zu melden, sofern man keinen Datenschutzbeauftragten hat.

  Die Meldepflicht entfällt aber, wenn unter 10 Personen mit der Verarbeitung der Daten beschäftigt sind, und die Betroffenen ihr Einverständnis erklärt haben, oder es der Abwicklung eines Vertrages oder ähnlichen Vertrauensverhältnisses dient.

- Hat die Firma einen Datenschutzbeauftragten, muß der natürlich entsprechend informiert werden.
- Falls man mit den Daten handelt, gilt die Meldepflicht an die Behörden auf jeden Fall.

Datenschutzrecht (14)

Inhalte der Meldepflicht:

- Verantwortliche Stelle (Firma, DV-Leiter)
- Zweck der Datenerhebung und -Verarbeitung
- Betroffene Personengruppen
- Daten oder Datenkategorien
- Empfänger, denen Daten mitgeteilt werden könnten
- Regelfristen für die Löschung der Daten
- geplante Datenübermittlung in Drittstaaten
- Maßnahmen zur Datensicherheit

Datenschutzrecht (15)

Vorabkontrolle:

- Während normalerweise diese Meldung ausreicht, ist bei besonders kritischen Fällen eine Vorabkontrolle notwendig:
  - ◇ bei besonders sensitiven Daten (Folie 12-30)
  - ◇ wenn die Verarbeitung der Daten dazu bestimmt ist, Persönlichkeit, Fähigkeiten, Leistung des Betroffenen zu bewerten.
- Zuständig ist der Datenschutzbeauftragte

  Kann sich in Zweifelsfällen an die Behörde wenden.

Datenschutzrecht (16)

Verbot von rein automatischen Entscheidungen:

- Es ist verboten, Entscheidungen über Menschen, die besonders negative Konsequenzen für sie haben können, rein automatisch (durch ein Computerprogramm) zu fällen.

  Mindestens müssen die Betroffenen darüber informiert werden, und die Möglichkeit bekommen, ihren Standpunkt darzustellen. Der Fall muß dann anschließend erneut betrachtet werden (durch einen Menschen).
- Wenn der Betroffene fragt, muß die grundlegende Logik der Entscheidung erklärt werden.

Datenschutzrecht (17)

Recht auf Information:

- Personen, über die Daten gespeichert werden, können die speichernde Stelle fragen,
  - ◇ welche Daten genau über sie gespeichert werden,
  - ◇ was die Quelle für diese Daten war,
  - ◇ zu welchem Zweck die Daten gespeichert sind,
  - ◇ an wen die Daten übermittelt wurden.
- Normalerweise müssen diese Fragen beantwortet werden, und in den meisten Fällen ohne Gebühr.

Datenschutzrecht (18)

Recht auf Information, Forts.:

- Die Polizei und ähnliche Behörden sind vom Recht auf Information natürlich ausgenommen.
- Man kann aber die Bundesdatenschutzbeauftragten bitten, die Daten zu prüfen.

  Weitere Ausnahmen: Wenn eine Firma mit Daten handelt, braucht sie die Frage, an wen die Daten übermittelt wurden, nicht zu beantworten (Geschäftsgeheimnis). Ggf. hat ein Richter zu entscheiden, was wichtiger ist. Firmen, die mit Daten handeln, können eine Gebühr für die Beantwortung der Fragen verlangen, wenn mit der Antwort Geld verdient werden kann. Die Gebühr kann aber niemals höher als die tatsächlichen Kosten sein. Gibt es Grund zur Annahme, daß die gespeicherten Daten inkorrekt sind, kann keine Gebühr verlangt werden.

Datenschutzrecht (19)

Recht auf Korrektur:

- Falsche Daten müssen berichtigt werden.

  Wenn die betroffene Person behauptet, die Daten seien falsch, und es läßt sich nicht feststellen, ob die Daten tatsächlich richtig oder falsch sind, müssen die Daten blockiert werden (können nicht mehr benutzt werden, sind als gelöscht markiert). Dies gilt nicht für Firmen, die mit Daten handeln, aber sie müssen den Daten eine Gegendarstellung der betroffenen Person beifügen und dürfen die Daten nicht ohne diese Gegendarstellung weitergeben.

- Falls die falschen Daten bereits weitergegeben wurden, muß der Empfänger informiert werden.

  Es sei denn, der Aufwand dafür wäre unverhältnismäßig hoch im Vergleich zum Schaden für die betroffene Person.

Datenschutzrecht (20)

Pflicht zur Löschung:

- Daten müssen gelöscht werden, wenn sie
  - ◇ illegal gespeichert wurden,
  - ◇ besonders sensitiv sind, und nicht bewiesen werden kann, daß sie korrekt sind,

    Siehe Folie 12-30, hier auch Information über illegale Aktivitäten.
  - ◇ nicht mehr für den ursprünglichen Zweck nötig sind.
- U.u. reicht es, sie als gelöscht zu markieren.

  Sie dürfen dann nicht mehr zugreifbar sein, können aber für im Gesetz beschriebene Zwecke wiederhergestellt werden.

Datenschutzrecht (21)

Technische Anforderungen:

- Um Mißbrauch der Daten zu verhindern, ist (in vernüftigen Grenzen) sicherzustellen, daß
  - ◇ nur zulässige Nutzer Zugriff auf die Rechner haben (Schutz vor Hackern),
  - ◇ diese Nutzer nur in den Grenzen ihrer Zugriffsrechte damit arbeiten können,
  - ◇ festgestellt werden kann, an wen (welche Firmen oder Behörden) Daten übermittelt wurden,

Datenschutzrecht (22)

Technische Anforderungen, Forts.:

- Es ist weiter sicherzustellen, daß
  - ◇ Daten bei der Übermittlung geschützt sind,

    Z.B. sollten Daten nur verschlüsselt über öffentliche Netze übertragen werden.
  - ◇ es möglich ist, festzustellen, wer die Daten eingegeben, geändert, gelöscht (?) hat,
  - ◇ die Daten vor Beschädigung/Verlust geschützt sind.

    Wenn z.B. festgehalten werden muß, an wen Daten übermittelt wurden, darf die Firma nicht einfach sagen können, “leider ist uns die Platte kaputt gegangen, auf der diese Daten standen.”

Datenschutzrecht (23)

Schadenersatz/Strafe:

- Wenn eine Firma/Behörde jemanden durch illegale oder inkorrekte Verarbeitung seiner/ihrer Daten schädigt, muß sie Schadenersatz zahlen.
- Dies gilt aber nicht, wenn sie einen angemessenen Standard an Sorgfalt erfüllt hat.

  Für Behörden gilt diese Einschränkung nicht, dafür ist die maximale Entschädigung begrenzt.
- Wenn dieses Gesetz vorsätzlich verletzt wurde, um damit Geld zu verdienen oder jemanden zu schädigen, ist die Maximalstrafe 2 Jahre Gefängnis.

Inhalt

GRANT Kommando (1)

- In SQL können Zugriffsrechte für Datenbankobjekte (Tabellen, Sichten, etc.) an andere Benutzer mit dem `GRANT` Kommando gegeben werden.
- `GRANT` war schon im SQL-86 Standard enthalten. Es hat die Form

  `GRANT ⟨Rechte⟩ ON ⟨Objekt⟩ TO ⟨Nutzer⟩`
- Das DBMS speichert diese Tripel im Systemkatalog ab, und prüft bei jedem Zugriff, ob der aktuelle Nutzer das entsprechende Recht hat.

  Es wird noch zusätzlich vermerkt, wer das Recht vergeben hat.

GRANT Kommando (2)

- Angenommen, die Tabelle `AUFGABEN` gehört dem Benutzer `BRASS`.
- Er kann Lese- und Einfüge-Rechte ("privileges") für diese Tabelle den Benutzern `MEIER` und `JUNG` mit folgendem Kommando geben:

  ```
  GRANT SELECT, INSERT ON AUFGABEN TO MEIER, JUNG
  ```
- Anschließend können `MEIER` und `JUNG` z.B.:
  - ◇ `SELECT * FROM BRASS.AUFGABEN`
  - ◇ `INSERT INTO BRASS.AUFGABEN VALUES (...)`

GRANT Kommando (3)

- `MEIER` und `JUNG` können die einmal eingegebenen Tabellenzeilen aber nicht ändern oder löschen.

  Auch nicht die Zeilen, die sie selbst eingegeben haben. Die Zeilen werden Bestandteil der Tabelle, die `BRASS` gehört. Wenn man nicht selbst etwas programmiert (z.B. eine Spalte mit `DEFAULT USER` (in Oracle), für die nicht explizit ein Wert angegeben werden kann: s.u.), ist nicht nachzuvollziehen, wer eine bestimmte Zeile eingegeben hat. Manche DBMS haben Erweiterungen für diesen Zweck.

- Man kann später z.B. dem Benutzer `MEIER` noch zusätzlich die fehlenden Rechte geben:

```
GRANT UPDATE, DELETE ON AUFGABEN TO MEIER
```

  Die bereits vorher vergebenen Rechte bleiben dabei bestehen.

GRANT Kommando (4)

Rechte für Tabellen/Sichten in SQL-92:

- `SELECT`: Lesezugriff (Anfragen an die Tabelle).

  In SQL Server kann man `SELECT` für einzelne Spalten vergeben. Das ist nicht im SQL-92 Standard und geht nicht in Oracle und DB2. Aber Sichten haben den gleichen Effekt.

- `INSERT`: Einfügen neuer Zeilen.
- `INSERT`($A_1, \ldots, A_n$): Nur für die Spalten A_i können Werte angegeben werden, die übrigen werden mit dem jeweiligen Defaultwert gefüllt (→ `CREATE TABLE`).

  `INSERT` nur für bestimmte Spalten ist Teil des SQL-92 Standards, aber nur in Oracle möglich (nicht in SQL Server und DB2). Mit Sichten kann man den gleichen Effekt erreichen.

GRANT Kommando (5)

Rechte für Tabellen/Sichten in SQL-92, Forts.:

- `UPDATE`: Änderung von Tabelleneinträgen.
- `UPDATE`($A_1, \ldots, A_n$): Nur Daten in den Spalten A_i können geändert werden.
- `DELETE`: Löschung von Tabellenzeilen.
- `REFERENCES`: Anlegen von Fremdschlüsseln, die auf diese Tabelle verweisen.

  Wenn Benutzer A in seiner Tabelle R auf die Tabelle S des Benutzers B verweist, aber B keine `DELETE`-Rechte an R bekommt, kann B referenzierte Zeilen aus seiner eigenen Tabelle S nicht mehr löschen. A kann auch prüfen, ob ein Schlüsselwert in S existiert.

GRANT Kommando (6)

Rechte für Tabellen/Sichten in SQL-92, Forts.:

- `REFERENCES`($A_1, \ldots, A_n$): Nur dieser Schlüssel darf in Fremdschlüsseln referenziert werden.

  Das ist nur wichtig, wenn eine Tabelle mehrere Schlüssel hat. Unterstützt in Oracle und DB2 (nicht in SQL Server).

Rechte für Domains/Zeichensätze, etc. in SQL-92:

- `USAGE`: Verwendung z.B. der Domain (benutzerdefinierter Datentyp) in `CREATE TABLE` Anweisungen.

  In keinem der drei DBMS (Oracle, DB2, SQL Server) unterstützt.

GRANT Kommando (7)

Weitere Rechte für Tabellen (nicht SQL-92 Standard):

- `ALTER`: Änderung des Schemas der Tabelle.

  Dieses Recht gibt es in Oracle and DB2, nicht SQL Server.

- `INDEX`: Anlegen eines Indexes auf der Tabelle.

  Dieses Recht gibt es in Oracle and DB2, nicht SQL Server.

Rechte für Prozeduren/Packages (nicht SQL-92):

- `EXECUTE`: Prozedur ausführen/aufrufen.

  Dieses Recht gibt es in allen drei DBMS.

- `BIND`: SQL Anweisungen in Paket neu optimieren.

  Nur in DB2.

GRANT Kommando (8)

Rechte für Schema Objekte (nur in DB2):

- `ALTERIN`: Änderung von Tabellen im Schema.

  Es geht dabei um die Änderung des Schemas dieser Tabellen, nicht der Daten (also z.B. Spalten hinzufügen, Constraints entfernen).

- `CREATEIN`: Anlegen von Tabellen im Schema.
- `DROPIN`: Löschen von Tabellen im Schema.

Rechte für Directory Objekte (nur Oracle):

- `READ`: Lesen von Dateien im Verzeichnis.

GRANT Kommando (9)

`GRANT ALL PRIVILEGES`:

- Statt einzelne Rechte aufzulisten, kann man auch alle Rechte weitergeben, die man selbst an einem Objekt (einer Tabelle) hat:

  `GRANT ALL PRIVILEGES ON ⟨Objekt⟩ TO ⟨Nutzer⟩`

`TO PUBLIC`:

- Anstatt alle Nutzer des Systems einzeln aufzulisten, kann man auch sagen

  `GRANT ⟨Rechte⟩ ON ⟨Objekt⟩ TO PUBLIC`

- Dies schließt zukünftige Nutzer mit ein.

GRANT Kommando (10)

`WITH GRANT OPTION`:

- Man kann einem Benutzer ein Recht auch mit der Möglichkeit geben, es an Dritte weiterzugeben.
- Dazu fügt man die Klausel "`WITH GRANT OPTION`" an das `GRANT`-Kommando an:

```
GRANT SELECT ON AUFGABEN TO MEIER
WITH GRANT OPTION
```

- `MEIER` kann nicht nur das Recht selbst, sondern auch die `GRANT OPTION` weitergeben.

GRANT Kommando (11)

Besitzer eines Objektes:

- Der Besitzer eines DB-Objektes (z.B. einer Tabelle), also der Nutzer, der das Objekt angelegt hat, hat alle Rechte (mit `GRANT OPTION`) an dem Objekt.
- Zunächst hat nur dieser Nutzer Rechte an dem Objekt. Andere Nutzer bekommen die Rechte nur mit expliziten `GRANT`s.

  Benutzer mit Administrator-Status können aber eventuell unabhängig von den mit `GRANT` vergebenen Rechten auf das Objekt zugreifen.
- Nur der Besitzer eines Objektes kann es wieder löschen (nicht in anderen Rechten enthalten).

GRANT Kommando (12)

`CONTROL`-Recht in DB2:

- Dies gibt alle Rechte an dem Objekt mit `GRANT OPTION`, und auch das Recht, das Objekt zu löschen.

  Es gibt kein `CONTROL`-Recht in Oracle und SQL Server. Aber es ist sehr ähnlich zum Besitzer-Status.
- Zunächst hat nur der Besitzer das `CONTROL`-Recht.

  Für Sichten bekommt der Besitzer das `CONTROL`-Recht nur, wenn er es auch auf den zugrundeliegenden Basistabellen und Sichten hat.
- Das `CONTROL`-Recht gibt es nur ohne `GRANT OPTION`. Nur ein Administrator kann es weitergeben.

  `GRANT CONTROL ON ...` braucht `SYSADM` oder `DBADM` Status.

Rücknahme von Rechten (1)

- Früher vergebene Rechte können mit einem Kommando zurückgenommen werden, des Syntax sehr ähnlich zum `GRANT` ist:

  `REVOKE ⟨Rechte⟩ ON ⟨Objekt⟩ FROM ⟨Nutzer⟩`
- `⟨Rechte⟩`: Durch Komma getrennte Liste einzelner Rechte oder “`ALL PRIVILEGES`”.
- `⟨Objekt⟩`: Name eines DB-Objektes (z.B. Tabelle).
- `⟨Nutzer⟩`: Durch Komma getrennte Liste einzelner Benutzernamen oder “`PUBLIC`”.

Rücknahme von Rechten (2)

- Beispiel:

  `REVOKE INSERT ON AUFGABEN FROM MEIER`

  Wenn `MEIER` vorher `SELECT` und `INSERT` Rechte für `AUFGABEN` hatte, hat er anschließend nur noch das `SELECT` Recht.
- Benutzer können nur Rechte zurücknehmen, die sie vorher selbst vergeben haben.

  Deshalb muß das DBMS in seinen internen Tabellen nicht nur Tripel Nutzer-Recht-Objekt speichern, sondern Quadrupel (A, P, O, B): Nutzer A hat das Recht P für Objekt O an Nutzer B vergeben.

Rücknahme von Rechten (3)

- Es ist nicht möglich, ein Recht an `PUBLIC` zu vergeben, und es dann von einem einzelnen Nutzer zurückzunehmen.

  Da sich `PUBLIC` auch auf zukünftige Nutzer bezieht, speichert das DBMS intern ab, daß das Recht an `PUBLIC` vergeben wurde, und nicht ein Tripel bzw. Quadrupel für jeden Nutzer einzeln.

- Man kann aber "`ALL PRIVILEGES`" vergeben, und die Rechte einzeln zurücknehmen.

  "`ALL PRIVILEGES`" bezieht sich auf die Rechte, die der Nutzer im Moment hat. Sie werden einzeln im Systemkatalog vermerkt.

Rücknahme von Rechten (4)

- Wenn eine Tabelle gelöscht wird, werden auch alle `GRANT`s für sie gelöscht.

  Im Systemkatalog gibt es also einen Fremdschlüssel mit "`ON DELETE CASCADE`" von der Rechte-Tabelle zur Objekt-Tabelle.

- Wenn die Tabelle später neu angelegt wird, hat wieder nur der Besitzer Zugriff auf sie.

Rücknahme von Rechten (5)

- Wenn A ein Recht "`WITH GRANT OPTION`" an B vergeben hat, und B es an C weitergegeben hat, und es A dann von B zurücknimmt, verliert es auch C:

$$\begin{tikzcd} A \arrow[r, "\times" marking] & B \arrow[r] & C \end{tikzcd}$$

- In SQL-92 benötigt dies `CASCADE`, siehe unten.
- Hat C das Recht aber auch auf einem anderen Weg bekommen (z.B. direkt von A), behält er es:

$$\begin{tikzcd} A \arrow[r, "\times" marking] \arrow[rr, bend right] & B \arrow[r] & C \end{tikzcd}$$

Rücknahme von Rechten (6)

- Das `REVOKE` Kommando versucht also die gleiche Situation wieder herzustellen, als hätte B das Recht nie gehabt.
- Wenn aber B das Recht genutzt hat, Tabellen zu ändern oder zu kopieren, können diese Aktionen natürlich nicht mehr ungeschehen gemacht werden.

  Wenn ein Benutzer einmal Lesezugriff für eine Tabelle gehabt hat, könnte er sich die Tabelle kopieren, oder ausdrucken, oder die Daten per EMail an einen anderen Rechner schicken.

Rücknahme von Rechten (7)

- In SQL-86 gab es kein `REVOKE` Kommando.
- In DB2 gibt nur das `CONTROL` Recht für ein DB-Objekt die Möglichkeit, Rechte für dieses Objekt zurückzunehmen.

  Das Recht ist dann auch weg, unabhängig von eventuell mehreren Pfaden (es sei denn, es wurde zusätzlich an `PUBLIC` oder eine Gruppe vergeben). Dies vereinfacht die Implementierung von DB2, aber es ist doch merkwürdig, daß Nutzer, die ein Recht mit `GRANT OPTION` bekommen haben, es zwar weitergeben können, aber nicht mehr zurücknehmen.

Rücknahme von Rechten (8)

- In SQL-92 und SQL Server muß man `CASCADE` hinzufügen, um Rechte rekursiv zurückzunehmen:

```
REVOKE INSERT ON AUFGABEN FROM MEIER CASCADE
```

  In SQL Server muß man `CASCADE` immer angeben, wenn man ein Recht zurücknimmt, das `WITH GRANT OPTION` vergeben wurde. In SQL-92 ist das nur nötig, wenn der Benutzer, dem das Recht entzogen wird, die `GRANT OPTION` tatsächlich benutzt hat, um das Recht weiterzugeben. In SQL-92 kann man auch `RESTRICT` statt `CASCADE` sagen, um das `REVOKE` nur auszuführen, wenn keine Rechte von anderen Benutzern davon abhängen. SQL Server versteht `RESTRICT` nicht.

- Oracle und DB2 verstehen `CASCADE/RESTRICT` nicht.

Rücknahme von Rechten (9)

- SQL-92 erlaubt auch

```
REVOKE GRANT OPTION FOR ...
```

  Dies wird nur in SQL Server verstanden, nicht in Oracle oder DB2. SQL Server verlangt allerdings, daß zusätzlich `CASCADE` angegeben wird.

- In Oracle muß man dem `REVOKE`-Befehl "`CASCADE CONSTRAINTS`" hinzufügen, wenn man ein `REFERENCES` Recht zurücknimmt.

  Sofern das Recht wirklich benutzt wurde, um Fremdschlüssel anzulegen. Diese Fremdschlüssel-Bedingungen werden dann gelöscht.

Inhalt

Zugriffsrechte für Sichten (1)

- Für Sichten kann man mit dem GRANT-Befehl anderen Benutzern Zugriffsrechte geben (genau wie für normale Tabellen).

  Wenn ein Benutzer lesend auf die Sicht zugreifen will, braucht er das SELECT-Recht für die Sicht. Manche Sichten sind auch änderbar, dann werden INSERT/UPDATE/DELETE-Anweisungen an die Sicht auf die Basistabellen umgerechnet (soweit möglich). Daher machen die betreffenden Recht ggf. auch für Sichten Sinn.

- Der andere Benutzer braucht keine Rechte an den Basistabellen, die der Sicht zugrunde liegen.

  Sichten werden ja gerade verwendet, um Zugriffe auf die Basistabellen einzuschränken.

Zugriffsrechte für Sichten (2)

- Es ist also ganz typisch, daß ein Benutzer z.B.
  ◇ das SELECT-Recht für die Sicht hat,
  ◇ aber nicht für die Basistabellen, die von der Sicht abgefragt werden.
- Dann kann er auf die Basistabellen nur über die Sicht zugreifen, und entsprechend
  ◇ nur einen Teil der Tabellen sehen (bestimmte Zeilen oder Spalten), bzw.
  ◇ nur aggregierte oder anonymisierte Daten.

Zugriffsrechte für Sichten (3)

- Aggregierte Daten lassen manchmal doch Rückschlüsse auf einzelne Personen zu:
  ◇ Möglicherweise ist in dieser Vorlesung nur eine Person, die Physik studiert. Die Durchschnittsnote nach Studiengang wäre dann problematisch.
  ◇ Die Punkte-DB zeigt die Anzahl der Teilnehmer mit höherer Punktzahl an. Ist sie 0, weiß man, daß jeder andere Teilnehmer, der diese Aufgabe abgegeben hat, nicht mehr Punkte hat.

Zugriffsrechte für Sichten (4)

- Man muß nicht unbedingt selbst Besitzer der verwendeten Basistabellen sein, um eine Sicht definieren zu können.
- Man braucht aber wenigstens das SELECT-Recht für die Basistabellen.
- Verliert man dieses Recht später (durch REVOKE), kann man die Sicht nicht mehr verwenden.

  Die Definition der Sicht wird aber nicht gelöscht. Die definierende Anfrage kann man immer noch im Systemkatalog nachschauen. Es kann ja auch sein, daß das REVOKE ein Fehler war, der gleich wieder durch einen GRANT-Befehl rückgängig gemacht wird. Es wäre ja schlimm, wenn dabei die Sicht gelöscht werden würde.

Zugriffsrechte für Sichten (5)

- Für den Zugriff auf eine Sicht
  - wird zunächst geprüft, ob der aktuelle Benutzer (der auf die Sicht zugreifen möchte) entsprechende Rechte an der Sicht hat.
  - Dann wird geprüft, ob der Benutzer, der die Sicht definiert hat, die notwendigen Rechte an den Basistabellen hat, und zwar "WITH GRANT OPTION".

    Definition einer Sicht und Weitergabe der Rechte daran hat ja dieselbe Wirkung wie die GRANT OPTION.
  - Falls der aktuelle Benutzer die Sicht selbst definiert hat, ist die GRANT OPTION nicht nötig.

Zugriffsrechte für Sichten (6)

- Obige Regeln gelten so in DB2 und Oracle.
- In SQL Server kann man Rechte an einer Sicht nur weitergeben, wenn man die zugrundeliegenden Tabellen besitzt.

  Genauer ist der GRANT-Befehl erfolgreich, aber die Zugriffe auf die Basistabellen werden in diesem Fall mit den mit den Rechten des Benutzers geprüft, der die Anfrage an die Sicht stellt, nicht des Benutzers, der die Sicht definiert hat (auch wenn dieser Benutzer die GRANT OPTION hat).
- In DB2 bekommt man das CONTROL-Recht an der Sicht nur, wenn man es für die Basistabellen besitzt.

Check Option (1)

- Angenommen, ein Tutor darf Punkte für alle Aufgabenarten außer der Endklausur eintragen:

```
CREATE VIEW TUTOR_BEW(SID, ATYP, ANR, PUNKTE) AS
SELECT SID, ATYP, ANR, PUNKTE
FROM   BEWERTUNGEN
WHERE  ATYP <> 'E'
```

- Überraschenderweise ist bei dieser Sichtdefinition möglich, Punkte für die Endklausur einzutragen:

```
INSERT INTO TUTOR_BEW VALUES (101, 'E', 1, 1000)
```

  Natürlich muß die Aufgabe in AUFGABEN eingetragen sein, sonst gibt es eine Fremdschlüssel-Verletzung. Das ist aber ein anderes Problem.

Check Option (2)

- Dieser Update wird übersetzt in:

```
INSERT INTO BEWERTUNGEN(SID, ATYP, ANR, PUNKTE)
            VALUES (101, 'E', 1, 1000)
```

  und anschließend ausgeführt.
- Das eingefügte Tupel wird in der Sicht nicht angezeigt, da es der WHERE-Bedingung aus der Sichtdefinition widerspricht.

  Es steht aber in der Basistabelle BEWERTUNGEN.
- Das Tupel kann über die Sicht nicht mehr geändert oder gelöscht werden (die Bedingung der Sicht wird ja jedem solchen Update angehängt).

Check Option (3)

- In der gleichen Weise ist möglich, daß ein Tupel nach einem UPDATE die Bedingung der Sicht verletzt:

```
UPDATE TUTOR_BEW
SET    ATYP = 'E'
WHERE  SID = 101 AND ATYP = 'H' AND ANR = 1
```

- Der Update bewirkt effektiv eine Löschung des Tupels aus der Sicht (es wird nicht mehr angezeigt).
- Auch hier wird der beabsichtigte Zugriffsschutz umgangen: Der Tutor kann beliebige Tupel einfügen.
- Lösung: WITH CHECK OPTION in Sichtdefinition.

Check Option (4)

- Die Sichtdefinition sieht jetzt so aus:

```
CREATE VIEW TUTOR_BEW(SID, ATYP, ANR, PUNKTE) AS
SELECT SID, ATYP, ANR, PUNKTE
FROM   BEWERTUNGEN
WHERE  ATYP <> 'E'
WITH   CHECK OPTION
```

- Jetzt wird geprüft, daß eingefügte oder geänderte Tupel auch die Bedingung der Sicht erfüllen.

  Also hinterher wirklich in der Sicht erscheinen.

- Verletzen sie die Bedingung, so wird die Einfügung / der Update nicht durchgeführt (Fehlermeldung).

Check Option (5)

- Warum ist die CHECK OPTION nicht Standard?
  - ◇ Performance-Nachteil: Zusätzlicher Test nötig.

    Oft nur minimale Verlangsamung. Je nach Sichtdefinition kann es aber auch nötig sein, dabei eine komplexe Unteranfrage auszuwerten. Immerhin kann man es vor dem Update auswerten, da die Unteranfrage sich nicht auf diese Tabelle beziehen kann.

  - ◇ Jetzt kann ein Update auf der Sicht nicht einfach in einen Update auf der Basistabelle übersetzt werden: Man bekommt einen Update mit vorgeschaltetem Test.
  - ◇ Nicht immer nötig (siehe nächste Folie).

Check Option (6)

- Falls der Tutor nur Zugriff auf Hausaufgaben haben soll, sieht die Situation ganz anders aus:

```
CREATE VIEW TUTOR_HA(SID, ANR, PUNKTE) AS
SELECT SID, ATYP, ANR, PUNKTE
FROM   BEWERTUNGEN
WHERE  ATYP = 'H'
```

  Einfügungen in die Sicht sind nur möglich, wenn 'H' Default-Wert für die Spalte ATYP in BEWERTUNGEN ist. Updates sind auf jeden Fall möglich.

- Die eingefügten bzw. geänderten Tupel können die Bedingung der Sicht nicht verletzen, da die verwendete Spalte ATYP in der Sicht ausgeblendet ist.