


帝塚山学院大学
 TEZUKAYAMA GAKUIN UNIVERSITY


データベース概論
トランザクション

中野秀男
情報メディア学科/ICTセンター長

1 データベース概論 トランザクション 2016/12/12


今日の話

- ▶ **今までの整理(まとめの紙に)**
 - ▶ データベースは集める、整理する、検索する
 - ▶ データ: 実体、データの性質: 属性、そしてデータの集合
 - ▶ いろいろなデータの集合があるので、それぞれに主キーを決めて、相互の関係は外部キーで関係をつける
 - ▶ 複数の利用者が同時に利用、権限制御などでSQLで統一
 - ▶ 高速に検索や処理のために便利なB-treeのデータ構造
 - ▶ レベルに応じて
 - ▶ Excel, Access, MySQL, PostgreSQL, SQLserver, Oracle
- ▶ 質問やコメントの回答
- ▶ テーブル設計: 今回は3つで、来週も
- ▶ トランザクション

2

 帝塚山学院大学
 TEZUKAYAMA GAKUIN UNIVERSITY


今までの整理(1)

- ▶ **データベースは集める、整理する、検索する**
 - ▶ 集めて整理する
 - ▶ 高速に処理したり、不具合がないように整理する
- ▶ **データ: 実体(entity), レコード**
 - ▶ 実体は属性を持っている, フィールド
- ▶ **実際の情報システムは複数のデータの集合を持っている**
 - ▶ 履修モデル
 - ▶ 眼鏡屋

3

 帝塚山学院大学
 TEZUKAYAMA GAKUIN UNIVERSITY


今までの整理(2)

- ▶ データの集合: 表, テーブル
 - ▶ いろいろなデータの集合があるので、
 - ▶ それぞれに主キーを決めて、
 - ▶ 相互の関係は外部キーで関係をつける
 - ▶ マスターテーブル(台帳): 基本になるテーブル
 - ▶ データテーブル: 日々更新されるデータ
- ▶ SQL
 - ▶ 複数の利用者が同時に利用、権限制御などでSQLで統一
 - ▶ ネットを通して利用できるように: サーバとクライアント(端末)
 - ▶ いろいろな言語から使えるように
 - ▶ PHP, Perl, Python, Ruby, C, C++, C#, Java
- ▶ 高速に検索や処理のために便利なB-treeのデータ構造

▶ 4 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


今までの整理(3): Excel

- ▶ 表はBook
- ▶ 行がレコード
- ▶ 列がフィールド
- ▶ だが行と列を変えたりできる
- ▶ 基本はセルで、セルの属性は数字、文字列、計算式
- ▶ 表レベルで良いとか、一人で利用する場合はExcelで十分
- ▶ VBAでさらに使いやすくなる
- ▶ 多くの利用者が同時に利用する場合はだめ
- ▶ 複数の表が相互に関係する場合もだめ

▶ 5 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY

今までの整理(4): Access

- ▶ 表はテーブル
- ▶ 行がレコード
- ▶ 列がフィールド
 - ▶ フィールドには数字、テキスト、日付などの属性を持たせる
 - ▶ Excelの場合は数字と文字列と計算式だけ
- ▶ 複数のテーブルが作れて、各テーブルに主キー
- ▶ 各テーブル間の関係のリンクができる
- ▶ フォームで便利な入力画面が作れる
- ▶ レポート機能でいろいろなレポートが出せる
- ▶ 多くの利用者が同時に利用する場合はだめ

▶ 6 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY

今までの整理(5): 本格的なデータベース

- ▶ MySQL, PostgreSQL, SQLserver, Oracle
- ▶ 表はテーブル
- ▶ 行がレコード
- ▶ 列がフィールド
- ▶ 複数のテーブルが作れて、各テーブルに主キー
- ▶ テーブルは外部キーを含ませることで他のテーブルと関連
- ▶ SQLをサポートしているので
 - ▶ テーブルの枠組みを定義するSQL文: create
 - ▶ データの操作をするSQL文: insert, delete, select
 - ▶ データのアクセス権限を指定するSQL文
 - ▶ 複数の言語から利用できる
 - ▶ ネットを通していろいろな場所からデータベースにアクセスできる
- ▶ 多くの利用者が同時に正しく利用できる

今までの整理(6): 本格的なデータベース

- ▶ MySQL, PostgreSQL
 - ▶ 両者ともオープンソースソフトウェア(OSS)
 - ▶ MySQLは商用ライセンスもある
 - ▶ PostgreSQLは完全にコミュニティベースのソフトウェア
- ▶ SQLserver
 - ▶ Windows ServerやWindowsで動くソフトウェア
- ▶ Oracle
 - ▶ 業界標準的なデータベース
- ▶ DB2

質問やコメント(1)

- ▶ だんだん難しくなってきたので復習を
- ▶ まとめはわかりやすかった
- ▶ 中野の予定
- ▶ 採点基準
- ▶ クエリで情報が簡単に引き出せる
- ▶ C-learningはSQLserver
- ▶ MySQLは初めて聞いた。OSや動作環境を聞きたい
- ▶ MySQLとPostgreSQLの違いがわからない
- ▶ MySQLを使っている企業

質問やコメント(2)

- ▶ 人工知能が知らないうちに進化している
- ▶ 立体でキャラが出ると一人でも寂しくなくていい
- ▶ ターミネータを思い出す
- ▶ ロボトミー
- ▶ 人の心や感情を読み取られるロボットはいらない
- ▶ 表情を読み取られるのは嫌
- ▶ レストランでお勧めがわかるロボット。個人情報の問題
- ▶ メニュー表がタッチパネル
- ▶ 血液の中にロボット。そんな日が来るか
- ▶ 機械の技術が進むと、逆に暮らしづらくなるのでは

▶ 10

データベース概論 トランザクション

2016/12/12



質問やコメント(3)

- ▶ iPhoneに人工知能が入っていると荷物が軽くなる
- ▶ 相談しながら仕事ができるのはいい
- ▶ 声の犯罪は怖い。見分け方は
- ▶ 自動運転の車が出ると免許はいらないか
- ▶ VRゲームに興味。いつ頃、フルタイプが。スマホ版は
- ▶ ルンバは思い通りに動かない
- ▶ ロボホンのプロモーション映像が良かった

▶ 11

データベース概論 トランザクション

2016/12/12



質問やコメント(4)

- ▶ ドローンでビデオ撮影
- ▶ ドローンは中国製がよく使われる
- ▶ ドローンの規則がなくなる日が
- ▶ 小型のドローンでも兵器になりそう
- ▶ ドローンは何の役に立つか。普通の人には関係なさそう
- ▶ 大学でドローンで何を記録しているのか。利用方法は
- ▶ 狭山キャンパスの点データ。3D
- ▶ 3Dがすごい
- ▶ 3Dデータを使ってUnityでゲームを
- ▶ 意外に綺麗に写っている

▶ 12


データベース概論 トランザクション

2016/12/12




テーブル設計(1)

- ▶ レコードとフィールド
 - ▶ レコードは行、entity
 - ▶ レコードの集まりがテーブルなのでテーブル設計が大事
 - ▶ レコードは複数のフィールドを持つ
 - ▶ フィールド名が固定されるとリレーショナルデータベース
 - 不確定とか後日追加変更だと半構造が便利
- ▶ レコードには主キーが
 - ▶ そのレコードをユニークに表すフィールドが主キー
 - ▶ またが複数のフィールド(例:履修科目テーブルの科目と教員)
- ▶ フィールドには外部キーもある
 - ▶ テーブル間の関係を示す外部キー
- ▶ これができると、次はどのアプリを使うか
 - ▶ Excel, Access, MySQL, PostgreSQL, SQLserver, Oracle

▶ 13 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(2):例1各都道府県データ

- ▶ 各都道府県データテーブル
- ▶ レコード:都道府県
- ▶ フィールド
 - ▶ 都道府県名
 - ▶ 知事の名前
 - ▶ 人口
 - ▶ 面積
 - ▶ 人口密度
 - ▶ 男性人口
 - ▶ 女性人口
 - ▶ 60歳以上
 - ▶ 花、歌など
- ▶ 拡張:未来予測

▶ 14 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(3):例2 レストラン(1)

- ▶ 店員マスター
- ▶ レコード:店員
- ▶ フィールド
 - ▶ 店員番号:主キー
 - ▶ 店員名
 - ▶ 店員種別:外部キー
 - ▶ 性別
 - ▶ 生年月日
 - ▶ 年齢
 - ▶ 勤務日時
 - ▶ 時給
 - ▶ テーブル

▶ 15 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(4):例2 レストラン(2)

- ▶ 料理マスターテーブル
 - ▶ レコード:料理
 - ▶ フィールド
 - ▶ 料理番号:主キー
 - ▶ 料理名
 - ▶ 料理種別:外部キー
 - ▶ 価格
- ▶ お客データテーブル(顧客は作らない)
 - ▶ レコード:お客
 - ▶ フィールド
 - ▶ オーダー番号:主キー
 - ▶ 性別
 - ▶ おおよその年齢
 - ▶ 料理:外部キー
- ▶ 半構造にしないと設計は難しそう

▶ 16 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(5):例3 履修モデル(1)

- ▶ 学生マスターテーブル
 - ▶ レコード:学生
 - ▶ フィールド
 - ▶ 学籍番号:主キー
 - ▶ 名前
 - ▶ 学科:外部キー
- ▶ 教員マスターテーブル
 - ▶ レコード:教員
 - ▶ フィールド
 - ▶ 教員番号:主キー
 - ▶ 名前
 - ▶ 学科:外部キー
- ▶ 科目マスターテーブル

▶ 17 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(6):例3 履修モデル(2)

- ▶ 科目マスターテーブル
 - ▶ レコード:科目
 - ▶ フィールド
 - ▶ 科目番号:主キー
 - ▶ 科目名
 - ▶ 担当教員:外部キー
 - ▶ 開講曜日と時限
- ▶ 学科マスターテーブル
 - ▶ レコード:学科
 - ▶ フィールド
 - ▶ 学科番号:主キー
 - ▶ 学科名
 - ▶ 学科長:外部キー
- ▶ 学部マスターテーブル

▶ 18 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


テーブル設計(7):例3 履修モデル(3)

- ▶ 履修データテーブル
 - ▶ レコード:履修科目
 - ▶ フィールド
 - ▶ 学生名
 - ▶ 学籍番号
 - ▶ 科目名
 - ▶ 科目番号
 - ▶ この場合は「学籍番号」+「科目番号」が主キー
 - ▶ 同じ科目名で複数の教員が担当したり、さらに複数時限にあるとか

19 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY


トランザクション

- ▶ トランザクション
 - ▶ データベースの状態を、整合のある状態から、別の整合性のある状態に変化させるデータ操作の集合
 - ▶ 整合のない状態とは
 - ▶ 例: 仕送り問題
 - ▶ トランザクションはコミットされるか、アボートされる
 - ▶ コミット: 正常終了
 - ▶ アボート: 異常終了
 - ▶ ロールバック
 - ▶ アボートした時に開始前の状態に戻す

20 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY

ACID特性

- ▶ 本格的なデータベースに要求される条件
 - ▶ A: atomicity(原子性)
 - ▶ トランザクションが更新後の状態になるか、まったく処理を行わなかった状態でおわるかのいずれか
 - ▶ C: consistency(一貫性)
 - ▶ 整合性がトランザクションの完了後も保たれる
 - ▶ I: isolation(分離性)
 - ▶ トランザクションが他のトランザクションに影響をあたえないように分離されて実行される
 - ▶ D: durability(持続性)
 - ▶ 一旦コミットされたトランザクションの更新は、その後の障害などで失われることはない

21 データベース概論 トランザクション 2016/12/12  帝塚山学院大学
TEZUKAYAMA GAKUIN UNIVERSITY

ロック

▶ **ロックによる同時実行制御**

- ▶ 操作対象のデータを施錠する
 - ▶ 例: 同時に同じデータの書き換えが起こった場合など
- ▶ 共有ロック: データ読み出しを行うためのロック
- ▶ 排他ロック: データの更新を行うためのロック
- ▶ ロックの粒度: どの範囲でロックするか
 - ▶ 例: 全体をロック、レコードをロック
