講義補助資料 No.7 (11/27)

168

リレーショナルデータベースの一貫性(1)

- **一貫性**（**integrity**）：保全性とも呼ばれる
  - データベースの内容がモデリングの対象となった実世界の事物およびそれらの間の関連を正しく表現していること
  - トランザクションの整合性(consistency)との関係

    基本的には同じものだが、整合性はトランザクションの前後で成り立つ性質であるのに対して、一貫性はデータベースの設計時に決める性質である
- **一貫性制約(integrity constraint)**：保全性制約、意味制約とも呼ばれる
  - データベースの一貫性を保証するための条件
- 一貫性制約の種類
  - ドメイン制約
  - キー制約、外部キー制約
  - 関数従属性、多値従属性

169

リレーショナルデータベースの一貫性(2)

- 一貫性制約は、**リレーションスキーマにおいて成り立つ**条件である

  つまり、ある瞬間（ある時刻のインスタンス）だけに成り立つのではなく、時刻に関係なく常に成り立つ
- 一貫性制約は、**データベースの更新のための条件**である

  一貫性制約は、リレーションの作成時に指定され、その制約に反するようなリレーションの更新（タップルの挿入／削除／修正）を許さないために使われる

170

ドメイン制約(domain integrity)

リレーションにおける各属性の**ドメインに対する制約**

CHECK制約と呼ばれることもある(例：最低賃金、年齢制限)

[SQLでの記述]

create domain ドメイン名 as データ型
　check (value 条件式)

create table 表名 (列名 ドメイン名, ..)

(例)

create domain salary as int
　check (value between 0 and 200)

create table emp (…, salary salary, …)

（注）条件式の記述としては、between以外に比較演算子を使った条件（例： salary >= 0）なども記述できる

171

補足：Oracleにおけるドメイン制約の記述

Oracleでは、create domainがないため、
create tableで以下のように制約を記述するか、
create typeを使うことによりドメイン制約を表現する

[例]

create table emp
　(…,
　salary number(6) **check (salary between 0 and 200),**
　…)

172

キー制約(key constraint)

- 主キー制約(primary key constraint)：

  空値でなく、かつ候補キーの条件を満たす

  実体完全性（entity integrity）とも呼ばれる
- 候補キー制約(candidate key constraint)：

  空値をとるか、または空値でないときは候補キーの条件を満たす

［参考］候補キーの条件

*R*をリレーションスキーマ**R**の任意のインスタンスとすると、次の2つの条件が成り立てば、属性集合*K*はリレーションスキーマ**R**の候補キーである

1. $(\forall\ t, t' \in R)(t[K]=t'[K] \rightarrow t=t')$
2. *K*から属性を一つでも欠落させると1．が成立しない

173

キー制約の記述

[SQLでの記述]

create table 表名
　(列名　ドメイン名 **not null**, ...) ←空値をとらない

create table 表名
　(列名　ドメイン名 **unique**, …) ←空値でないときは、この属性が同じ値をとるタップルは1個しかない（候補キー制約に対応）

create table 表名
　(列名　ドメイン名 **primary key**, ...) ←この属性が主キーである（空値をとらず、かつこの属性が同じ値をとるタップルは1個しかない。主キー制約に対応。）

講義補助資料 No.7 (11/27)

外部キー(foreign key)と外部キー制約
174

- あるリレーション(自分自身を含む)の主キーを参照できる属性
- リレーションスキーマ**R**, **S**の任意の時刻*τ*でのインスタンス*Rτ*, *Sτ*とすると、**R**のある属性集合*H*について以下の条件が成り立つとき、*H*を**S**の<u>外部キー</u>と呼ぶ

<u>外部キー制約</u>(foreign key constraint)

参照整合性(**referential integrity**)とも呼ばれる

1. *Rτ*の任意のタップル*t*において、*t*[*H*]は空値か、そうでなければ、
2. *t*[*H*]= *s*[*K*] となる*Sτ*のタップル*s*が存在する(但し*K*は**S**の主キー)

(注意)**R**と**S**は同じリレーションスキーマであっても良い

外部キーの記述
175

[SQLでの記述]

create table 表1

(列1 ドメイン名, ...

foreign key(列1) **references** 表2)

- 列1は表2の**外部キー**であることを宣言
- 表1にタップルが挿入または修正されたときに、タップルの列1に表2の主キーにない値が入っているとエラーとなる

キーのまとめ
176

- **超キー (super key)**
  - ある属性集合の値により、リレーションの全属性集合の値が決まる(＝タップルが一意に定まる)

  条件1 (∀ *t*, *t'* ∈ *R*)(*t*[*K*]=*t'*[K] ⇒ *t*=*t'*)
- **候補キー (candidate key)**
  - 超キーのうち最小のもの(属性集合を一つでも除くと条件1が満たされないもの)
- **主キー (primary key)**
  - あるリレーションの候補キーのうち、空値を取らず、主要なもの(どれが主要かはデータベースの設計者が決める)
- **外部キー (foreign key)**
  - <u>他のリレーションで、主キー</u>となっているもの

関数従属性(functional dependency)
177

- リレーションスキーマ**R**の属性集合X, Y (XとYは互いに素である必要はない)について次の条件を満たすとき、「**XからYへの関数従属性が存在する**」といい、 X →Y と書く

  (∀ t,t' ∈ *R*)(t[X]=t'[X] ⇒ t[Y]=t'[Y]).

  ここで*R*はリレーションスキーマ**R**の任意のインスタンスである

[参考] X →Yを、「XはYを関数的に決定する」または「YはXに関数従属している」とも言う

関数従属性の意義
178

- 関数従属性はリレーションスキーマにおいて成り立つ(いつでも常に成り立つ)
- **一貫性制約の一つ**である
  - 定義されている関数従属性に反するようなリレーションの更新(タップルの挿入、削除、修正)は許されない
- リレーションの更新時の異状を抑制するための、第一正規形より高次の正規形を規定する基礎となる(リレーショナルデータベースの設計の箇所で説明)

完全関数従属性 (full functional dependency)
179

- リレーションスキーマRの属性集合X, Y (XとYは互いに素である必要はない)について、次の二つの条件がともに成立するとき、「XからYへの<u>完全</u>関数従属性が存在する」(または「YはXに完全関数従属している」)という

1. X →Y,
2. **Xのいかなる<u>真部分集合</u>X'に対しても X' → Yが成立しない**.

講義補助資料 No.7 (11/27)

180

自明な関数従属性 (trivial functional dependency)

- 次の関数従属性は常に成立するため、<u>**自明な関数従属性**</u>と呼ばれる。

1. Y ⊆ Xのとき、X → Y,
2. X → φ,
3. X → X.

181

関数従属性と超キー（super key）

- *H*がリレーションスキーマ**R**の超キーであることは、関数従属性を使って、

$$H \rightarrow \overline{H} \quad (\overline{H} = \Omega_R - H)$$

と表すことができる

［参考］候補キーと超キー

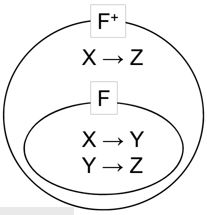
*R*をリレーションスキーマ**R**の任意のインスタンスとすると 、次の2つの条件が成り立てば、属性集合*K*はリレーションスキーマ**R**の候補キーである（1. しか成り立たないときは**超キー**と呼ぶ）

1. (∀ *t*, *t'* ∈ *R*)(*t*[*K*]=*t'*[*K*] ⇒ *t*=*t'*)
2. *K*から属性を一つでも欠落させると1．が成立しない

182

関数従属性の閉包

- リレーションスキーマ**R**の関数従属性の集合Fが与えられたとき、Fから導出されるすべての関数従属性の集合を**Fの閉包**(closure)と呼び、F^+で表す。



183

関数従属性の公理系

- <u>**関数従属性の公理系**</u>（アームストロングの公理系とも呼ばれる）は、リレーションスキーマ**R**の属性集合X, Y, Zに対して次のように表される。
  - **反射律**: もしY ⊆ Xならば、X → Y（自明な関数従属性の一つ），
  - **添加律**: もしX→YでZ ⊆ Ω_Rならば、X ∪ Z → Y ∪ Z（∪は和集合演算），
  - **推移律**: もしX → YかつY → Zならば、X → Z.
- 関数従属性の公理系は、**健全(sound)**でかつ**完全(complete)**であることが知られている
  - **R**で成り立つ任意の関数従属性は、与えられた関数従属性の集合に対して、上記の公理系の導出規則を有限回適用することにより得ることができる

184

関数従属性の導出

- 関数従属性の公理系を使うことにより、与えられた関数従属性の集合から、別の関数従属性を導出することができる
- 導出の例:

（与えられている関数従属性）

｛学番，科目｝→得点、　｛科目，得点｝→判定

（導出したい関数従属性）｛学番，科目｝→判定

（導出過程）

｛学番，科目｝→得点　　①‥所与

｛学番，科目｝→｛得点，科目｝　②‥①に添加律を適用

｛科目，得点｝→判定　　③‥所与

｛学番，科目｝→判定　　④‥②と③に推移律を適用

185

多値従属性 (multivalued dependency)

- リレーションスキーマ**R**の属性集合X, Y (XとYは互いに素である必要はない)について次の条件を満たすとき，「XからYへの<u>**多値従属性**</u>が存在する」といい、

X →→ Yと書く

(∀t,t'∈*R*)

(**t**[X]=**t'**[X]⇒(**t**[X∪Y],**t'**[Z])∈R∧(**t'**[X∪Y],**t**[Z])∈R).

ここで*R*はリレーションスキーマ**R**の任意のインスタンスであり、Z = Ω_R - (X∪Y)である

[参考] X→→Yを、「XはYを多値に決定する」または「Yは Xに多値に従属している」ともいう

講義補助資料 No.7 (11/27)

186

多値従属性の直感的な説明

講習
（例：講習1）

講師グループ
（例：{講師A，講師B}）

受講者グループ
（例：{受講者1，受講者2}）

意味的には、**集合への関数従属性**
（集合の要素が一つなら関数従属性）

講習	講師グループ	受講者グループ
講習1	{講師A, 講師B}	{受講者1, 受講者2}
講習2	{講師C, 講師D}	{受講者3, 受講者4}

講習	講師グループ	受講者グループ
講習1	講師A	受講者1
講習1	講師A	受講者2
講習1	講師B	受講者1
講習1	講師B	受講者2

なぜ「関数従属性」として扱えないのか？

187

多値従属性が生じる原因は正規化にある

世帯（正規化前）

世帯	家族
世帯1	{大阪太郎、大阪春子、大阪一太郎}
世帯2	{神戸次郎、神戸夏子、神戸小次郎}

dom（家族）=2$^{\mathrm{dom}（構成員）}$

世帯（正規化後）

世帯	構成員
世帯1	大阪太郎
世帯1	大阪春子
世帯1	大阪一太郎
世帯2	神戸次郎
世帯2	神戸夏子
世帯2	神戸小次郎

巾集合(power set)とは、ある集合のすべての部分集合からなる集合
例えば、ドメインD={1,2,3}であったとすると、Dの巾集合2^D={{}, {1}, {2}, {3}, {1,2}, {2,3}, {1,3}, {1,2,3}}

188

多値従属性の性質

リレーションスキーマ**R**の属性集合X, Yの間に多値従属性X→→ Yが成り立つとき、

- 任意のタップルt, t'(∈R)において、t[X]=t'[X]であれば、
  (t [X∪Y], t [Z]), (t'[X∪Y], t'[Z]) （元々の組合せ）、
  (t [X∪Y], t'[Z]), (t'[X∪Y], t [Z]) （多値従属性の定義）
  の組合せ（すべての組合せ）がR中に含まれている
- **R**の任意のインスタンス*R*における射影$\pi_X(R)$と$\pi_Y(R)$の間の関係と、射影$\pi_X(R)$と$\pi_Z(R)$（ $Z = \Omega_R$ -(X∪Y)）の間の関係は<u>**直交(orthogonal)している （独立である）**</u>
- リレーションスキーマ**R**の属性集合X, Yにおいて、X→→ Yが成り立つとき、$Z = \Omega_R$ -(X∪Y)であるZについてもX →→Zが成り立つ（このことを示すため、X→→ Y | Zと書くことがある）

189

多値従属性の例

リレーションスキーマ**講習会**（講習，講師，受講者）で、次の多値従属性が成り立つ場合を考える

講習 →→ 講師 | 受講者

- 講習と講師の関係と、講習と受講者の関係は**直交している**

（一つの講習を2人の講師A,Bで分担して担当する場合、任意の受講者Xが
講師Aの講習を受けて入れば、必ず、
講師Bの講習も受けている）

190

自明な多値従属性 (trivial multivalued dependency)

- 次の多値従属性は常に成立するため**自明な多値従属性**と呼ばれる

X∪Y = Ω_Rのとき、X →→Y | φ,
Y ⊆ Xのとき、X→→ Y.

191

関数従属性と多値従属性

- **すべての関数従属性は多値従属性である**
  X, Y⊆Ω_Rである任意の属性集合XとYについて，X → Yであれば X →→ Yが成り立つ。
  この意味で、多値従属性は関数従属性の一般化と見なすことができる。

講義補助資料 No.7 (11/27)

関数従属性と多値従属性(2)

- **すべての関数従属性は多値従属性であることの証明**

関数従属性の定義：

(∀ t,t’ ∈ R)(t[X]=t’[X] ⇒ t[Y]=t‘[Y]). ①

多値従属性の定義：

(∀t,t’∈R)
(t[X]=t’[X]⇒(t[X∪Y],t’[Z])∈R∧(t’[X∪Y],t[Z])∈R). ②

証明：①からt[X]=t’[X] ⇒ t[X∪Y]=t’[X∪Y]が成り立つ。
このとき、(t[X∪Y],t’[Z])= (t’[X∪Y],t’[Z])=t’であり、
(t’[X∪Y],t[Z])= (t[X∪Y],t[Z])=tであるから、t,t’∈Rであることから②が成り立つ。

多値従属性の公理系

- リレーションスキーマ**R**上の関数従属性の集合をF、多値従属性の集合をMとすると、F∪Mから導出されるすべての関数従属性または多値従属性の集合(F∪Mの閉包)を得るための完全でかつ健全な次のような公理系が知られている。

以下ではX, Y, Zをリレーションスキーマ**R**の属性集合とする。

- **関数従属性の反射律**：もしY⊆Xならば、X → Y,
- **関数従属性の添加律**：もしX→YでZ ⊆Ω_Rならば、X∪Z→Y∪Z,
- **関数従属性の推移律**：もしX→YかつY→Zならば、X→Z,
- **多値従属性の相補律**：もしX→→YでZ=Ω_R - (X∪Y)ならば、X →→Z,
- **多値従属性の添加律**：もしX→→YでZ⊆W⊆Ω_Rならば、X∪W →→ Y∪Z,
- **多値従属性の推移律**：もしX→→YかつY→→Zならば、X→→(Z - Y),
- **模写律**：もしX→Yならば、X→→Y,
- **合体律**：もしX→→YでZ⊆Yであるとき、W∩Y=φ であるWについてW→Zであれば、X→Z.

多値従属性の導出の例

リレーションスキーマ**講習会**（講習，講師，設備，受講者）で、次の関数従属性と多値従属性が成り立つとする

講習 →→ {講師，設備} ｜ 受講者 ①‥所与
受講者 → 設備 ②‥所与

このとき，設備⊆{講師，設備},受講者∩{講師，設備}=φ であるので，①，②と合体律より、

講習 → 設備

が成り立つ