Errata for
Database System Concepts 3rd Ed.

Avi Silberschatz Henry F. Korth S. Sudarshan

July 2, 2000

This document contains errata for Database System Concepts, 3rd edition. For each chapter there is a list of errata (if any) in the 3rd printing; these are also present in the 1st and 2nd printing. There is also a list of errata from the 1st and 2nd printing that have been fixed in the 3rd printing. If you have the third or later printing you can disregard these errata.

To identify the version of the book which you have, look at the ISBN – the ISBN of the 3rd printing is 0-07-031086-6, while the previous two printings have an ISBN of 0-07-044756-X.

Chapter 1

Errata in 3rd and earlier Printings

- **Page 5, Para. 4**: In "This code defines a new record called *customer* with three fields.", change "three" to "four".
- **Page 8, Para. 4**: Change "An an illustration ..." to "As an illustration ...".
- **Page 10, Para. 1**: Change ".. 321-12-3123, lives on Main in Harrison .." to ".. 192-83-7465, lives on Alma in Palo Alto ..".

Chapter 2

Errata in 3rd and earlier Printings

- **Page 41, Para. 6**: In "When more than one specialization is formed on an entity set, a particular entity may belong to both of the specializations."

  change "both of the" to "multiple".
- **Page 57, Para. 4**: In "The *savings-account* and *checking-account* relations corresponding to these tables both have *balance* as the primary key",
  change *balance* to *account-number*.
- **Exercise 2.2**: Delete "of Exercise 2.2"
- **Exercise 2.13**: In figure 2.24(a), replace the box around R by a diamond.

Chapter 3

Errata in 3rd and earlier Printings

- **Page 69, Figure 3.7**: Change the line from *account* to *account-branch*, as well as the line from *loan* to *loan-branch* to double lines. Also add the following sentence to the end of the first paragraph:

  > Note that the tables for *account-branch* and *loan-branch* have been combined into the tables for *account* and *loan* respectively, since the participation of *account* and *loan* in the corresponding relationships is total, as indicated by the double lines in Figure 3.7.

- **Page 84, Figure 3.22**: Drop the row (Williams, Perryridge) from the table.

- **Page 89, Para. 1**: Change the first expression in this page to:

$$\{t \mid \exists\, r \in \mathit{customer}\ (r[\mathit{customer\text{-}name}] = t[\mathit{customer\text{-}name}]) \wedge$$
$$(\ \forall\, u \in \mathit{branch}\ (u[\mathit{branch\text{-}city}] = \text{“Brooklyn”} \Rightarrow$$
$$\exists\, s \in \mathit{depositor}\ (t[\mathit{customer\text{-}name}] = s[\mathit{customer\text{-}name}]$$
$$\wedge\, \exists\, w \in \mathit{account}\ (w[\mathit{account\text{-}number}] = s[\mathit{account\text{-}number}]$$
$$\wedge\, w[\mathit{branch\text{-}name}] = u[\mathit{branch\text{-}name}])))\}$$

  The difference from the earlier version is additional condition in the first line of the new expression.

- **Page 92, Para. 5**: Change the expression for the query “Find the names of all customers who have an account at all the branches located in Brooklyn.” to

$$\{< c > \mid \exists\, n\ (< c, n > \in \mathit{customer}) \wedge$$
$$\forall\, x, y, z\ (< x, y, z > \in \mathit{branch} \wedge y = \text{“Brooklyn”} \Rightarrow$$
$$\exists\, a, b\ (< x, a, b > \in \mathit{account} \wedge\ < c, a > \in \mathit{depositor}))\}$$

  The difference from the earlier version is additional condition in the first line of the new expression.

- **Page 99, Figure 3.34**: In the column *sum of salary*, change 5500 to 5300, and 7900 to 8100.

- **Page 100, Para. 4**: Change

  – Delete all of Smith's accounts.

$$\mathit{account} \leftarrow \mathit{account} - \sigma_{\mathit{customer\text{-}name} = \text{“Smith”}}\ (\mathit{account})$$

  to

  – Delete all of Smith's account records.

$$\mathit{depositor} \leftarrow \mathit{depositor} - \sigma_{\mathit{customer\text{-}name} = \text{“Smith”}}\ (\mathit{depositor})$$

- **Page 102, Para. 2**: Change
  “Consider a person who needs to know a customer's loan number ...” to:
  “Consider a person who needs to know a customer's loan number and branch name, ...”
  and change
  $\Pi_{\mathit{customer\text{-}name},\ \mathit{loan\text{-}number}}\ (\mathit{borrower} \bowtie \mathit{loan})$
  to $\Pi_{\mathit{customer\text{-}name},\ \mathit{loan\text{-}number},\ \mathit{branch\text{-}name}}\ (\mathit{borrower} \bowtie \mathit{loan})$

Errata in 1st and 2nd Printings only

- **Exercise 3.15**: In

  3.15 Using the bank example, write relational-algebra queries to find out how many accounts are held by more than two customers in the following ways:

  (*a*) using an aggregate function.

  (*b*) without using any aggregate functions.

  change “find out how many accounts are held” to “find the accounts held”

Chapter 4

Errata in 3rd and earlier Printings

- **Page 132, Para. 1**: Change "... in which case all tuples in P are deleted." to "... in which case all tuples in r are deleted.".

Errata in 1st and 2nd Printings only

- **Page 130, Section 4.7 (Derived Relations)**:
  In *both* the SQL queries in this section change "*depositor*" to "*account*"

- **Page 132, Para. 3**: Change the item
  Delete all of Smith's account records.
  delete from *depositor*
  where *customer-name* = "Smith"
  to
  Delete all account records in the Perryridge branch.
  delete from *account*
  where *branch-name* = "Perryridge"
  (Note: The incorrect version was based on a schema that was used in the 2nd edition.)

- **Page 132, Para. 3**: Change the item starting
  Delete all accounts at every branch located in Perryridge.
  ⋮
  to
  Delete all account records at every branch located in Needham.
  delete from *account*
  where *branch-name* **in** (**select** *branch-name*
  from *branch*
  where *branch-city* = "Needham")
  The preceding **delete** request first finds all branches in Needham, and then deletes all *account* tuples pertaining to those branches.
  (Note: The incorrect version used "Perryridge" as a city name, whereas it is a branch name in our examples. Note that the **delete** request does not delete account number information in the *depositor* relation.)

- **Page 146, Para. 5**: Change the query starting

  EXEC SQL
  declare *c* **cursor for**
  ⋮
  to

  EXEC SQL
  declare *c* **cursor for**
  select *customer-name, customer-city*
  from *depositor, customer, account*
  where *depositor.customer-name* = *customer.customer-name* **and**
  account.account-number = *depositor.account-number* **and**
  account.balance > :*amount*
  END-EXEC
  (Note: The incorrect version was based on a schema that was used in the 2nd edition.)

- **Exercise 4.10**: In
  The operation returns $result_i$, where i is the first of $pred_1, pred_2, \ldots, pred_b$
  change $pred_b$ to $pred_n$.

Chapter 6

Errata in 3rd and earlier Printings

- **Page 200, Para. 3**: In last line of the first assertion, change *loan.branch-name* to *account.branch-name*, and in the previous line, change **sum**(*amount*) to **sum**(*balance*).

Chapter 7

Errata in 3rd and earlier Printings

- **Figure 7.1, 7.3 and 7.4**: In all these figures, in the rows containing "Johnson", change "L-23" to "L-18".

Errata in 1st and 2nd Printings only

- **Page 229, Para. 5 (Section 7.3.3 Third Normal Form)**: In the sentence

  > The fact that each relation schema R_i is in 3NF follows directly from our requirement that the set F of functional dependencies be in canonical form (Section 6.5.4).

  The above fact does not follow "directly". For a proof, see Ullman [1988].

- **Page 236, Para. 4**: Change

  > It is identical to the BCNF decomposition algorithm of Figure 7.6, except for the use of multivalued, instead of functional, dependencies.

  to

  > It is identical to the BCNF decomposition algorithm of Figure 7.6, except for the use of multivalued, instead of functional, dependencies, and the use of the "restriction of D^+ to R_i" (defined below).

- **Figure 7.12**:

  1. Change

     compute D^+;

     to

     compute D^+; Given schema R_i, let D_i denote the restriction of D^+ to R_i

  2. In

     if (there is a schema R_i in *result* that is not in 4NF)

     change "in 4NF" to "in 4NF w.r.t. D_i"

  3. In

     on R_i such that $\alpha \rightarrow R_i$ is not in D^+, and $\alpha \cap \beta = \emptyset$;

     change D^+ to D_i.

- **Exercise 7.6**: In

  > Show that it is possible to ensure that a decomposition into 3NF is a lossless-join decomposition by guaranteeing that at least one schema contains a candidate key for the schema being decomposed.

  change "a decomposition into 3NF" to "a dependency preserving decomposition into 3NF".

Chapter 9

- **Exercise 9.1**: Assume that the attribute names of attributes of type **setof***Children*, **setof***Skills*, and **setof***Exams*, are respectively *ChildrenSet*, *SkillsSet* and *ExamsSet*.

- **Exercise 9.3**: Change

  Consider the schemas for the table *person*, and the tables *student* and *teacher*, which were created under *person*, in Section 9.2.2.

  to

  Consider the schemas for the table *people*, and the tables *students* and *teachers*, which were created under *people*, in Section 9.2.2.

(Note: The SQL-3 specifications have been undergoing continuous development, and several changes have happened since the chapter was written. Look on our web home page for an update on SQL-3.)

Chapter 10

Errata in 3rd and earlier Printings

- **Page 328, Figure 10.22 and 10.23**: Change "679.34.28000" to "679.34278", and "519.56.84000" to "519.56850'. (Note that here and below, only a volume and page number are needed.)

- **Page 329, Para. 4**: Change "679.342.78" to "679.34278", and "519.568.50" to "519.56850'.

- **Page 330, Para. 2 and 3**: Change "679.342.78" to "679.34278", and "519.568.50" to "519.56850'.

- **Page 336, Exercise 10.23**: Change "679.342.78" to "679.34278".

Errata in 1st and 2nd Printings only

- **Page 294, Para. 1**: Change

  "Cache memory is small; its use is managed by the operating system."

  to

  "Cache memory is small; its use is managed by the hardware."

- **Page 305, Para. 2 (RAID Level 1**:) Change:

  RAID level 1 refers to disk mirroring, which we discussed earlier.

  to

  RAID level 1 refers to disk mirroring with block striping.

- **Page 305, Para. 6 (RAID Level 4**:) Change:

  RAID Level 4, or block-interleaved parity organization, stores blocks just like in regular disks, without striping them across disks, but keeps a parity block on a separate disk for corresponding blocks from N other disks.

  to

  RAID Level 4, or block-interleaved parity organization, uses block level striping, and in addition keeps a parity block on a separate disk for corresponding blocks from N other disks.

- **Page 307, Para. 3**: Change:

  If there are more disks in an array, data-transfer rates are higher but, overall, fewer I/O operations can be performed per second.

  to

  If there are more disks in an array, data-transfer rates are higher, but the system would be more expensive.

(Note: Overall, fewer I/O operations can be performed per second. for RAID levels 2 and 3 (bit striping), but not for levels 4 and 5 (block striping).)

- **Page 332, Para. 3**: Change

  > Large objects must be stored in a contiguous sequence of bytes when they are brought into memory; if an object is bigger than a page, contiguous pages of the buffer pool must be allocated to store it, which makes buffer management more difficult.

  to

  > Large objects may need to be stored in a contiguous sequence of bytes when they are brought into memory; in that case, if an object is bigger than a page, contiguous pages of the buffer pool must be allocated to store it, which makes buffer management more difficult.

Chapter 11

Errata in 3rd and earlier Printings

- **Page 356, Para. 5**: Change

  > The pointers P_i are the tree pointers that we used also for B^+-trees. The pointers B_i in the nonleaf nodes are the bucket or file-record pointers.

  to

  > Leaf nodes are the same as in B^+-trees. In nonleaf nodes, the pointers P_i are the tree pointers that we used also for B^+-trees, while the pointers B_i are bucket or file-record pointers.

- **Page 358, Para. 7**: Change "loyokup" to "lookup".

Errata in 1st and 2nd Printings only

- **Figure 11.7**: Flip the pointers to Brighton and Downtown — the pointer from the left of Brighton in the leaf node should point to the Brighton record in the *account* file, and the pointer from the left of Downtown should point to Downtown.

Chapter 12

Errata in 3rd and earlier Printings

- **Page 391, Para. 5**: Change

  > So, as was the case for the index on *branch-name*, the index for *balance* has a depth of 2, and two block accesses are required to read the first index block. In the worst case, there are 50 leaf nodes, one-half of which must be accessed. This accessing leads to 25 more block reads."

  to

  > The index for *balance* thus has a depth of 3, and three block accesses are required to read the first index block. In the worst case, there are 50 leaf nodes, one-half of which must be accessed. This accessing leads to 24 more block reads.

- **Page 409-410, Section 12.6.7**: Change all occurrences of *loan* to *account*, *loan-number* to *account-number*, and *branch* to *account* in this section.

Errata in 1st and 2nd Printings only

- **Page 393, Para. 2**: Change "constitute the first scan" to "be tested first".
- **Page 422, Para. 3**: Change

  The selection operation distributes over the union, intersection, and set-difference operations.

  $$\sigma_P(E_1 - E_2) = \sigma_P(E_1) - E_2 = \sigma_P(E_1) - \sigma_P(E_2)$$

  Similarly, for union and intersection, the preceding equivalence, with set difference ($-$) replaced with either one of $\cup$ or $\cap$, also holds.

  to

  The selection operation distributes over the union, intersection, and set-difference operations.

  $$\sigma_P(E_1 - E_2) = \sigma_P(E_1) - \sigma_P(E_2)$$

  The above equivalence, with $-$ replaced by either $\cup$ or $\cap$, also holds. Further,

  $$\sigma_P(E_1 - E_2) = \sigma_P(E_1) - E_2$$

  The above equivalence, with $-$ replaced by $\cap$, also holds.

- **Exercise 12.15**: Change
  "Suppose that a B^+-tree index on *branch-name* is available"
  to
  "Suppose that a B^+-tree index on *branch-city* is available"
- **Exercise 12.16**: Change the exercise to:

12.16 Suppose that a B^+-tree index on (*branch-name, branch-city*) is available on relation *branch*. What would be the best way to handle the following selection?

$$\sigma_{(branch\text{-}city < \text{“Brooklyn”}) \wedge (assets < 5000) \wedge (branch\text{-}name = \text{“Downtown”})}(branch)$$

(Note: The index is now on a different set of attributes, and a mistake in the expression has been corrected.)

- **Exercise 12.19(c)**: Change this part of the question to:
  $\sigma_\theta(E_1 \ltimes E_2) = \sigma_\theta(E_1) \ltimes E_2$ where θ uses only attributes from E_1.
  (The earlier version said "uses only attributes from E_2", which is wrong.)
- **Exercise 12.21(c)**: : Change part (c) of the question to:
  - (*c*) In the above expression, if *max* is replaced by *min* would the expressions be equivalent?

  (The incorrect version in the book requires you to show that the expressions are not equivalent.)

Chapter 14

Errata in 1st and 2nd Printings only

- **Page 500, Section 14.7.3**: Replace the index-locking protocol by the following:
  - Every relation must have at least one index. Access to a relation must be made only through one of the indices on the relation.
  - A transaction T_i that performs a lookup (either a range lookup or a point lookup) must lock all the index buckets that it accesses in S-mode.
  - A transaction T_i may not insert a tuple t_i into a relation r without updating all indices to r. T_i must perform a lookup on every index to find all index buckets that could have possibly contained a pointer to tuple t_i, had it existed already, and obtain locks in X-mode on all these index buckets. T_i must also obtain locks in X-mode on all index buckets that it modifies.
  - The rules of the two-phase locking protocol must be observed.

Chapter 15

- **Page 536, Para. 5**: In the paragraph beginning "Therefore, when the insertion ..":
  1. Change <O_i, **operation-end**, U> to <T_i, O_j, **operation-end**, U>.
  2. Change "O_i denotes an identifier for the operation" to "O_j denotes a unique identifier for (the instance of) the operation".
- **Page 536, Para. 6**: Change
  "<O_i, **operation-begin**>, where O_i is the identifier"
  to
  "<T_i, O_j, **operation-begin**>, where O_j is the unique identifier".
- **Page 537, Para. 1**: Change
  1. "Whenever a log record <O_i, **operation-end**, U> log record"
     to
     "Whenever a log record <T_i, O_j, **operation-end**, U>".
  2. "Further, **operation-begin** and an **operation-end** log records are generated just like during normal operation execution."
     to
     "But instead of generating a log record $< T_i, O_j$, **operation-end**, $U >$, a log record $< T_i, O_j$, **operation-abort**> is generated."
  3. "the log record <O_i, **operation-begin**>"
     to
     "the log record <T_i, O_j, **operation-begin**>".
- **Page 537, Para. 2**: Add the following paragraph just before the paragraph beginning with "If failures occur while a logical operation is in progress...":

  If a record $< T_i, O_j$, **operation-abort**> is found, all preceding records are skipped till we find the record $< T_i, O_j$, **operation-begin**>. These preceding log records must be skipped to prevent multiple rollback of the same operation, in case there had been a crash during an earlier rollback, and the transaction had already been partly rolled back. When the transaction T_i has been rolled back, a record <T_i **abort**> is added to the log.
- **Page 537, Section 15.9.3**: Exchange steps **2** and **3** of the checkpoint procedure. (Otherwise a system crash during checkpointing can cause an error.)
- **Exercise 15.4**: Change "prior to committing of that transaction." to "prior to data updated by the transaction being written to disk."

Chapter 16

- **Page 552, Section 16.3.2 (last item)**: In
  > In contrast, in a mesh a component may be $\sqrt{n} - 1$ links away from some of the other components (or $\sqrt{n}/2$ links away, if the mesh interconnection wraps around at the edges of the grid).

  change $\sqrt{n} - 1$ to $2(\sqrt{n} - 1)$, and $\sqrt{n}/2$ to $\sqrt{n}$.

Chapter 18

- **Page 610, Section 18.5.2.2**: (Case where log contains a <**precommit** T> record, but no <**abort** T> or <**commit** T> record.) Change:

  > If C_i fails to respond within a prescribed interval, the site executes the coordinator-failure protocol.

to

If C_i fails to respond within a prescribed interval, the site waits for a coordinator to be chosen and inform it of the decision.

- **Page 620, Section 18.8.2 (Fully Distributed Approach)**: Change

  Each site maintains its own local wait-for graph. A local wait-for graph differs from the wait-for graph described previously in that we add one additional node T_{ex} to the graph. T_{ex} represents transactions that are external to the local site. An arc $T_i \rightarrow T_{\text{ex}}$ exists in the graph if T_i is waiting for a data item in another site that is being held by *any* transaction. Similarly, an arc $T_{\text{ex}} \rightarrow T_j$ exists in the graph if a transaction at another site is waiting to acquire a resource currently being held by T_j in this local site.

  to

  We assume that a transaction runs at a single site, and makes requests to other sites for accessing non-local data. Each site maintains its own local wait-for graph in the normal fashion; thus an edge $T_i \rightarrow T_j$ if T_i is waiting on a lock held by T_j. Note that T_i and T_j may be non-local transactions. In addition, the local wait-for graph has one additional node T_{ex}. An arc $T_i \rightarrow T_{\text{ex}}$ exists in the graph at site S_k if either (a) T_i is executing at site S_k, and is waiting for a reply to a request made on another site, or (b) T_i is non-local to site S_k, and a lock has been granted to T_i at S_k. Similarly, an arc $T_{\text{ex}} \rightarrow T_i$ exists in the graph at a site S_k if either (a) T_i is non-local to site S_k, and is waiting on a lock for data at site S_k, or (b) T_i is local to site S_k, and has accessed data from an external site.

Chapter 19

No bugs reported yet.
(Note: The SQL-3 standardization effort has seen considerable progress in recent times. Look on our web home page for some of the recent developments.)

Chapter 20

- **Page 674, Para. 2**: In

  Systems based on the process-per-server model, such as the original version of the IBM CICS TP monitor and file servers such as Novel's NetWare, successfully provided high transaction rates with limited resources.

  change "process-per-server" to "single-server".

Chapter 21

Errata in 3rd and earlier Printings

- **Page 734, Section 21.10.1**: In the section heading, and in the first part of the section, change "Universal Resource Locators" to "Uniform Resource Locator"

Errata in 1st and 2nd Printings only

- **Exercise 21.17**: In

  (Hint: Use the example from Exercise 21.16, with the assumption that documents A and B are available on both mobile computers A and B, and take into account the possibility that a document may be read with it being updated.)

  change (a) "documents A and B" to "documents 1 and 2", and (b) "with it being updated" to "without its being updated".