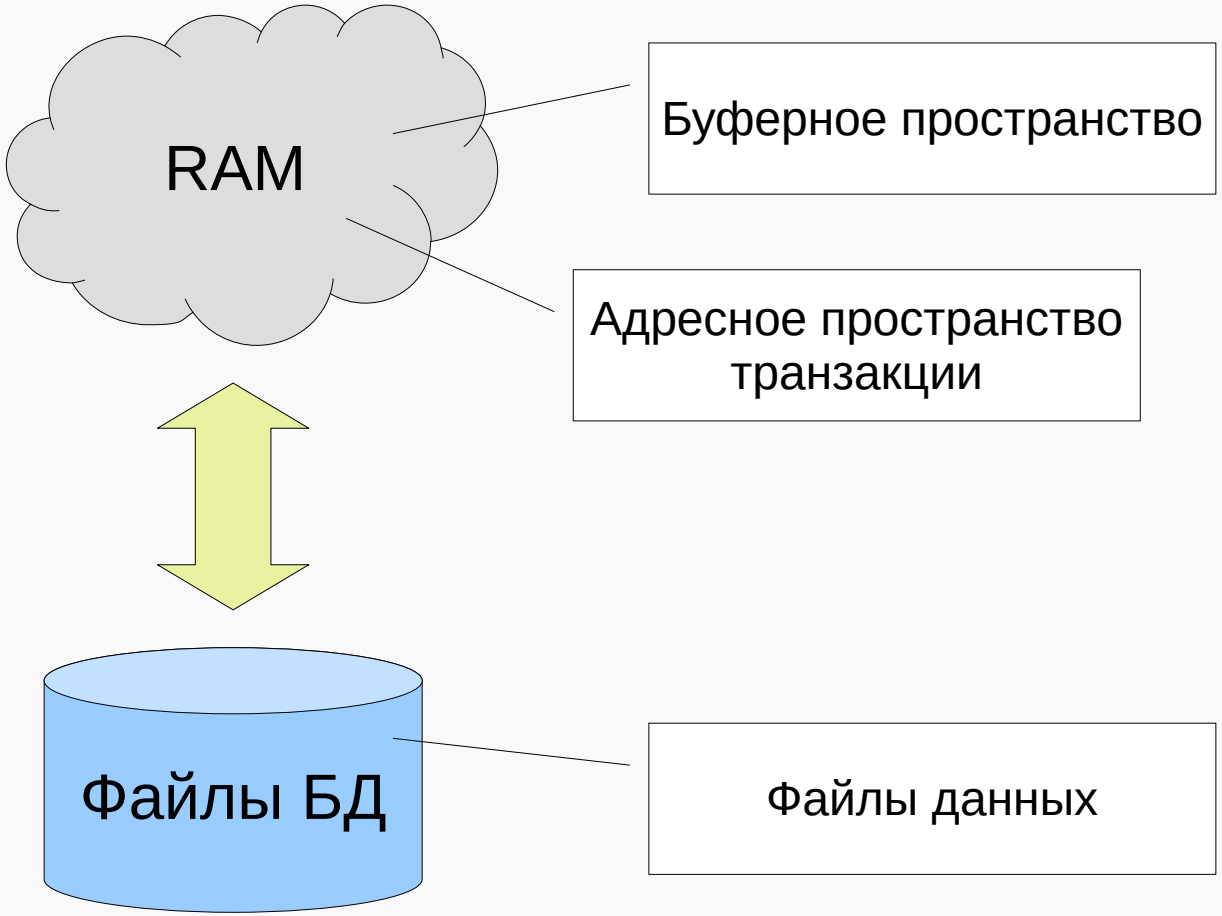
Лекция 5

1. Выполнение операций и восстановление данных

ИТМО ВТ

Память



ИТМО ВТ

Операции для синхронизации различных пространств памяти

- M ← D(Obj) — загрузить страницу, содержащую Obj, с диска в буферное пространство.
- D ← M(Obj) — сохранить страницу, содержащую Obj, на диск из буфера ОП.
- T(v, Tx) ← M(Obj) — копировать Obj в переменную v транзакции Tx:
  - если Obj не в M, то сначала M ← D(Obj).
- M(Obj) ← T(v, Tx) — копировать значение v из транзакции Tx в Obj:
  - если Obj не в M, то сначала M ← D(Obj).
- WriteLog — сохранить лог на диск из соответствующего буфера.

ИТМО ВТ

Пример

SELECT * FROM ROOMS;

Room

Room_ID	Address	Room_Num	Size
1	Kronverksky, 49	374	20
2	Kronverksky, 49	375	25

ИТМО ВТ

Пример

Задача: переименовать аудитории

Room

Room_ID	Address	**Room_Num**	Size
1	Kronverksky, 49	**374**	20
2	Kronverksky, 49	**375**	25

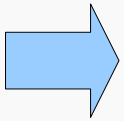
```
BEGIN;
UPDATE ROOM SET Room_Num = 1331
WHERE Room_Num = 374;
UPDATE ROOM SET Room_Num = 1330
WHERE Room_Num = 375;
COMMIT;
```

Пример

*Выполнение операции UPDATE сильно упрощено

BEGIN;

```
UPDATE ROOM
SET Room_Num = 1331
WHERE Room_Num = 374;
```

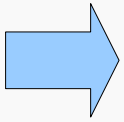


$T(v_{Room}, Tx) \leftarrow M(Room_{374})$;

$V_{Room}(Room\_Num) = 1331$;

$M(Room_{374}) \leftarrow T(v_{Room}, Tx)$;

```
UPDATE ROOM
SET Room_Num = 1330
WHERE Room_Num = 375;
```



$T(v_{Room}, Tx) \leftarrow M(Room_{375})$;

$V_{Room}(Room\_Num) = 1330$;

$M(Room_{375}) \leftarrow T(v_{Room}, Tx)$;

$D \leftarrow M(Room_{374}), M(Room_{375})$

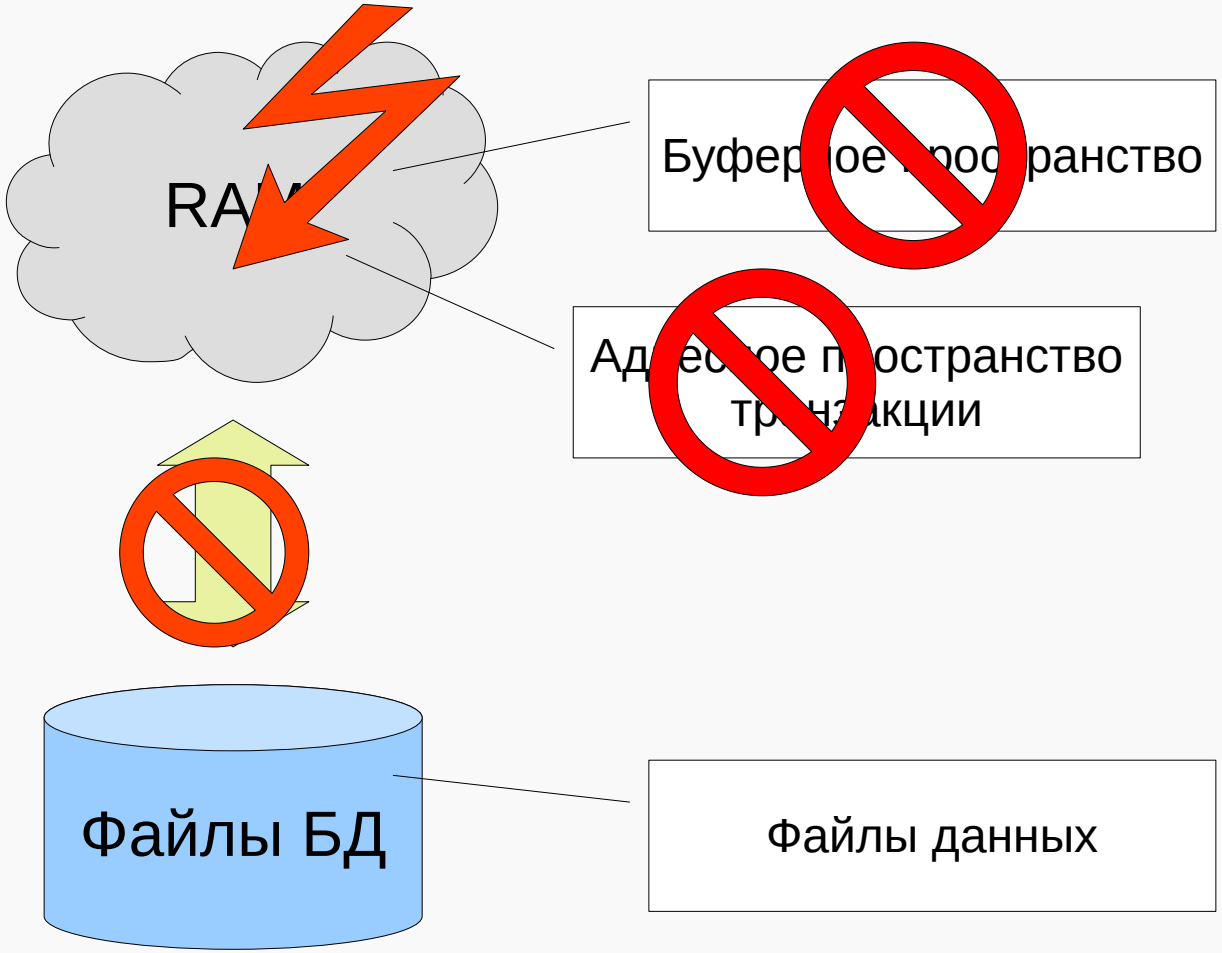
COMMIT;

ИТМО ВТ

Пример

Операция	Пам. тр.	Буферное пространство	Диск
$T(v_R, Tx) \leftarrow M(R_{374})$	374	R_{374}=374, R_{375}= -	R_{374}=374, R_{375}=375
V_R(Room_Num) = 1331;	1331	R_{374}=374, R_{375}= -	R_{374}=374, R_{375}=375
$M(R_{374}) \leftarrow T(v_R, Tx)$;	1331	R_{374}=1331, R_{375}= -	R_{374}=374, R_{375}=375
$T(v_R, Tx) \leftarrow M(R_{375})$;	375	R_{374}=1331, R_{375}=375	R_{374}=374, R_{375}=375
V_R(Room_Num) = 1330;	1330	R_{374}=1331, R_{375}=375	R_{374}=374, R_{375}=375
$M(R_{375}) \leftarrow T(v_R, Tx)$;	1330	R_{374}=1331, R_{375}=1330	R_{374}=374, R_{375}=375
$D \leftarrow M(R_{374})$	1330	R_{374}=1331, R_{375}=1330	R_{374}=1331, R_{375}=375
$D \leftarrow M(R_{375})$	1330	R_{374}=1331, R_{375}=1330	R_{374}=1331, R_{375}=1330

«Незапланированный» отказ работы экземпляра



ИТМО ВТ

Пример

Операция	Пам. тр.	Буферное пространство	Диск
$T(v_R, Tx) \leftarrow M(R_{374})$	374	R_{374}=374, R_{375}= -	R_{374}=374, R_{375}=375
V_R(Room_Num) = 1331;	1331	R_{374}=374, R_{375}= -	R_{374}=374, R_{375}=375
$M(R_{374}) \leftarrow T(v_R, Tx)$;	1331	R_{374}=1331, R_{375}= -	R_{374}=374, R_{375}=375
$T(v_R, Tx) \leftarrow M(R_{375})$;	375	R_{374}=1331, R_{375}=375	R_{374}=374, R_{375}=375
V_R(Room_Num) = 1330;	1330	R_{374}=1331, R_{375}=375	R_{374}=374, R_{375}=375
$M(R_{375}) \leftarrow T(v_R, Tx)$;	1330	R_{374}=1331, R_{375}=1330	R_{374}=374, R_{375}=375
$D \leftarrow M(R_{374})$	1330	R_{374}=1331, R_{375}=1330	R_{374}=1331, R_{375}=375
$D \leftarrow M(R_{375})$			

Если система «упадет» в данный момент — проблема!

2. UNDO-журнал

ИТМО ВТ

Логирование

- Для предотвращения ситуаций, связанных с появлением несогласованных данных можно использовать — **журнал** (лог).

- СУБД сохраняет информацию об изменениях и фиксациях данных в журнале:
  - первоначально запись сохраняется в соответствующем буфере (буфере журнала);
  - буфер журнала синхронизируется с файлами журнала на диске (WriteLog);

ИТМО ВТ

Журнал отмены

- Один из вариантов ведения журнала — журнал отмены, UNDO LOG

  Возможные записи в журнале отмены:

- BEGIN Tx -- начало транзакции;
- Tx: R, old_val -- изменение: транзакция Tx
  -- переписывает значение
  -- old_val в R;
- COMMIT Tx -- успешное завершение транзакции;
- ABORT Tx -- преждевременное завершение транз-ии;
- START CHKn: T_n, …, T_m -- начало создания контр. точки
- END CHKn -- завершение создания контр. точки

ИТМО ВТ

Журнал отмены

Правила записи в UNDO LOG:

- Запись изменения данных (Tx: R, old_val) записывается в журнал (на диск) ДО сохранения обновленного значения на диске.
- COMMIT Tx должен быть добавлен в журнал (на диск) **ПОСЛЕ** обновления всех файлов данных, связанных с изменениями данной транзакции (после синхронизации буферного пространства данных с файлами данных).

ИТМО ВТ

Пример

Операция	Диск	Журнал
		BEGIN T1;
$T(v_R, Tx) \leftarrow M(R_{374})$	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1331;	R_{374}=374, R_{375}=375	
$M(R_{374}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{374}, 374
$T(v_R, Tx) \leftarrow M(R_{375})$;	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1330;	R_{374}=374, R_{375}=375	
$M(R_{375}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{375}, 375
WriteLog		-- записи на диске
$D \leftarrow M(R_{374})$	R_{374}=1331, R_{375}=375	
$D \leftarrow M(R_{375})$	R_{374}=1331, R_{375}=1330	
		COMMIT T1
WriteLog		-- COMMIT на диске

ИТМО ВТ

Использование журнала отмены для восстановления данных (1)

Действия при восстановлении данных (1):

- СУБД сканирует журнал от новых записей до старых. Отдельно фиксируются транзакции:
  - Группа 1: для которых есть запись COMMIT Tx;
  - Группа 2: транзакции для которых есть запись ABORT Tx, незавершенные транзакции.
- Когда в журнале встречается запись изменения (Tx: R, old_val), анализируется Tx. Если Tx:
  - из группы 1 действия не предпринимаются.
  - из группы 2: для R восстанавливается old_val (происходит отмена изменений).

Использование журнала отмены для восстановления данных (2)

Действия при восстановлении данных (2):

- СУБД записывает ABORT Tx для каждой незавершенной транзакции из группы 2.
- Запускается операция WriteLog, чтобы записи вида ABORT Tx оказались в журнале.

Пример (UNDO LOG)

Операция	Диск	Журнал
		BEGIN T1;
$T(v_R, Tx) \leftarrow M(R_{374})$	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1331;	R_{374}=374, R_{375}=375	
$M(R_{374}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{374}, 374
$T(v_R, Tx) \leftarrow M(R_{375})$;	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1330;	R_{374}=374, R_{375}=375	
$M(R_{375}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{375}, 375
WriteLog		-- записи на диске
$D \leftarrow M(R_{374})$	R_{374}=1331, R_{375}=375	

Произошел сбой!

Пример (UNDO LOG)

Операция	Диск	Журнал
		BEGIN T1;
$T(v_R, Tx) \leftarrow M(R_{374})$	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1331;	R_{374}=374, R_{375}=375	
$M(R_{374}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{374}, 374
$T(v_R, Tx) \leftarrow M(R_{375})$;	R_{374}=374, R_{375}=375	
V_R(Room_Num) = 1330;	R_{374}=374, R_{375}=375	
$M(R_{375}) \leftarrow T(v_R, Tx)$;	R_{374}=374, R_{375}=375	T1: R_{375}, 375
WriteLog		-- записи на диске
$D \leftarrow M(R_{374})$	R_{374}=1331, R_{375}=375	

Возвращаем первоначальное значение

Возвращаем первоначальное значение

ИТМО ВТ

Контрольные точки

- Для восстановления данных нужна какая-то стартовая точка, чтобы не анализировать транзакции с самого начала работы с БД.
- Контрольная точка (checkpoint) — точка синхронизации, во время которой происходит синхронизация данных из буферов с соответствующими файлами на диске.

ИТМО ВТ

Контрольные точки

Создание контрольной точки не одномоментно:

1) В журнал добавляется START CHKn: T_n, …, T_m, фиксируется на диске через WriteLog.
2) Ожидание завершения транзакций, которые работали в момент создания контрольной точки (T_n, …, T_m). В это время могут начинаться другие транзакции.
3) В журнал добавляется END CHKn, фиксируется на диске через WriteLog.
4) Записи перед START CHKn, у которой есть END CHKn, могут быть удалены.

UNDO LOG и контрольные точки

Действия при восстановлении данных (3):

- СУБД сканирует журнал и встречает запись END CHKn: Tn, …, Tm (а не START CHKn):
  - Значит установка контрольной точки n завершена;
  - Сканирование журнала до записи START CHKn, так как незавершенные транзакции могут быть только после записи START CHKn:
    - среди транзакций, которые начали работу после начала установки контрольной точки.
  - Остальные действия как раньше.

UNDO LOG и контрольные точки

Действия при восстановлении данных (4):

- СУБД сканирует журнал и встречает запись START CHKn: Tn, …, Tm (а не END CHKn):
  - Значит установка контрольной точки n не завершена (проблема во время установки);
  - Сканирование журнала до первой записи из самой ранней транзакции среди Tn, …, Tm:
    - такая запись будет до START CHKn;
  - Остальные действия как раньше.

ИТМО ВТ

Недостатки UNDO LOG

- Нельзя завершить транзакцию (зафиксировать COMMIT в журнале) до записи изменений данных в файлы данных.
- Много операций ввода-вывода с диском, так как надо синхронизировать данные для каждой транзакции.