



Data Science e Tecnologie per le Basi di Dati

Classificazione (1 punto, penalità -15% per risposta sbagliata)

Data la matrice di confusione in figura, quale delle seguenti asserzioni è verificata?

		Predicted	
		T	F
Actual	T	80	0
	F	10	90

- (a) La precisione della classe **F** è **90/100**, l'accuratezza è **170/180**
- (b) La precisione della classe **T** è **80/90**, il richiamo della classe **F** è **90/100** ✓
- (c) Nessuna delle risposte è corretta
- (d) Il richiamo della classe **F** è **90/100**, l'accuratezza è **90/180**
- (e) La precisione della classe **T** è **80/90**, il richiamo della classe **F** è **10/100**
- (f) La precisione della classe **F** è **90/100**, il richiamo della classe **T** è **80/90**

Clustering (1 punto, penalità -15% per risposta sbagliata)

La metrica **MAX** (o complete linkage) nel clustering gerarchico agglomerativo prevede che:

- (a) I cluster ottenuti siano molto sensibili al rumore
- (b) Nessuna delle risposte è corretta ✓
- (c) Un cluster **C** verrà unito ad un singolo punto **p** se la massima distanza tra **p** ed i punti di **C** è la maggiore nella matrice delle distanze
- (d) I primi due punti ad unirsi nel dendrogramma sono quelli più distanti tra loro
- (e) Due cluster **C₁**, **C₂** vengono uniti se esiste una coppia di punti **p₁ ∈ C₁**, **p₂ ∈ C₂** la cui distanza è la maggiore nella matrice delle distanze
- (f) Un cluster **C** verrà unito ad un singolo punto **p** se la distanza tra **p** e **C** è quella massima nella matrice delle distanze

Accesso concorrente (1 punto, penalità -15% per risposta sbagliata)

Secondo le definizioni del locking gerarchico per la gestione della concorrenza in un DBMS:

- (a) Se lo stato di un nodo è **Intention of Shared Lock (ISL)**, posso richiedere un **Intention of Exclusive Lock (IXL)** sullo stesso nodo ✓
- (b) Se lo stato di un nodo è **Intention of Exclusive Lock (IXL)**, posso richiedere uno **Shared Lock** sullo stesso nodo
- (c) Per poter richiedere un **Intention of Shared Lock (ISL)** non è necessario richiedere permessi sul nodo padre
- (d) Se lo stato di un nodo è **Intention of Exclusive Lock (IXL)**, non posso richiedere un **Intention of Shared Lock (ISL)** sullo stesso nodo
- (e) Posso richiedere un **Intention of Exclusive Lock (IXL)** su un nodo figlio dopo aver ottenuto uno **Shared Lock (SL)** sul nodo padre
- (f) Per poter richiedere un **Exclusive Lock (XL)** su un nodo figlio è necessario richiedere un **Exclusive Lock (XL)** sul nodo padre

Recovery (1 punto, penalità -15% per risposta sbagliata)

È data la seguente sequenza di operazioni all'interno di un file di log:

$B(T_1)$ $I_1(o_1)$ $B(T_2)$ $CK(T_1, T_2)$ $B(T_3)$ $U_3(o_2)$ $D_1(o_4)$ $Commit(T_1)$ $U_2(o_1)$ $Abort(T_2)$ $I_3(o_5)$ **FAILURE**

Notazione:

- T_n = Id della transazione n
- $B(T_n)$ = Begin di T_n
- CK = checkpoint
- $U_n(o_x)$ = update eseguito da T_n sull'oggetto o_x ; stessa notazione per I (insert) e D (delete)

Quali operazioni vengono eseguite per un **warm restart**?

- (a) Redo di T_2 , undo di T_1, T_3
- (b) Redo di T_1, T_2 , undo di T_3
- (c) Redo di T_3 , undo di T_1, T_2
- (d) Redo di T_1, T_3 , undo di T_2
- (e) Redo di T_1 , undo di T_2, T_3 ✓
- (f) Redo di T_2, T_3 , undo di T_1

Cardinalità (2 punti, penalità -15% per risposta sbagliata)

SMART-DEVICE(SerialId, Nome, Brand, Tipologia, Prezzo)
 FEATURE(FeatureId, Nome, Categoria)
 DEVICE-HA-FEATURE(SerialId, FeatureId, Valore, UnitaDiMisura, DisplayValue)
 USER(Username, Nome, Cognome, DataNascita, Address, Citta', Nazione)
 ACQUISTO(Timestamp, Username, SerialId, CostoTotale, NumeroDiPezzi)

Assumere le seguenti cardinalità:

- $\text{card}(\text{SMART-DEVICE}) = 5 \cdot 10^6$ tuple
- valori distinti di Tipologia = 50
- $\text{card}(\text{FEATURE}) = 10^2$ tuple
- valori distinti di Categoria = 10
- $\text{card}(\text{DEVICE-HA-FEATURE}) = 10^8$ tuple
- $\text{card}(\text{USER}) = 10^6$ tuple
- valori distinti di Nazione = 100
- $\text{card}(\text{ACQUISTO}) = 5 \cdot 10^7$ tuple
- $\text{MIN}(\text{DATE}(\text{Timestamp})) = 1/1/2018, \text{MAX}(\text{DATE}(\text{Timestamp})) = 31/12/2019$

Inoltre, assumere il seguente fattore di riduzione per la condizione di group by:

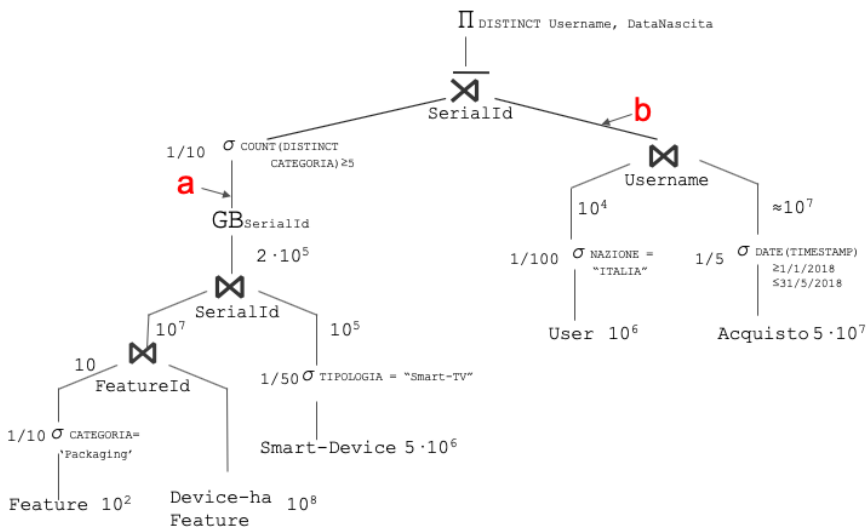
HAVING COUNT(DISTINCT F.Categoria) >= 5 $\approx 1/10$

Si consideri la seguente query:

```
select DISTINCT Username, DataNascita
from USER U, ACQUISTO A
where U.Username=A.Username and Nazione='Italia'
and DATE(Timestamp) >= 1/1/2018 and DATE(Timestamp) <= 31/5/2018
and SerialId NOT IN (select SerialId
from SMART-DEVICE D, FEATURE F, DEVICE-HA-FEATURE DHF
where D.SerialId=DHF.SerialId and F.FeatureId=DHF.FeatureId
and F.Categoria='Packaging'
D.Tipologia = 'SmartTV'
group by SerialId
having COUNT(DISTINCT F.Categoria)>=5)
```

La figura sottostante rappresenta il query tree per la query precedente.

Specificare la cardinalità di ogni ramo indicato dai caratteri in rosso (a, b) nella figura sottostante.



- (a) a: 10^5 , b: $5 \cdot 10^5$
- (b) a: $5 \cdot 10^5$, b: $5 \cdot 10^5$

Indici (2 punti, penalità -15% per ogni risposta sbagliata)

SMART-DEVICE(SerialId, Nome, Brand, Tipologia, Prezzo)
 FEATURE(FeatureId, Nome, Categoria)
 DEVICE-HA-FEATURE(SerialId, FeatureId, Valore, UnitàDiMisura, DisplayValue)
 USER(Username, Nome, Cognome, DataNascita, Address, Città, Nazione)
 ACQUISTO(Timestamp, Username, SerialId, CostoTotale, NumeroDiPezzi)

Assumere le seguenti cardinalità:

- card(SMART-DEVICE) = $5 \cdot 10^6$ tuple
- valori distinti di Tipologia = 50
- card(FEATURE) = 10^2 tuple
- valori distinti di Categoria = 10
- card(DEVICE-HA-FEATURE) = 10^8 tuple
- card(USER) = 10^6 tuple
- valori distinti di Nazione = 100
- card(ACQUISTO) = $5 \cdot 10^7$ tuple
- MIN(DATE(Timestamp)) = 1/1/2018, MAX(DATE(Timestamp)) = 31/12/2019

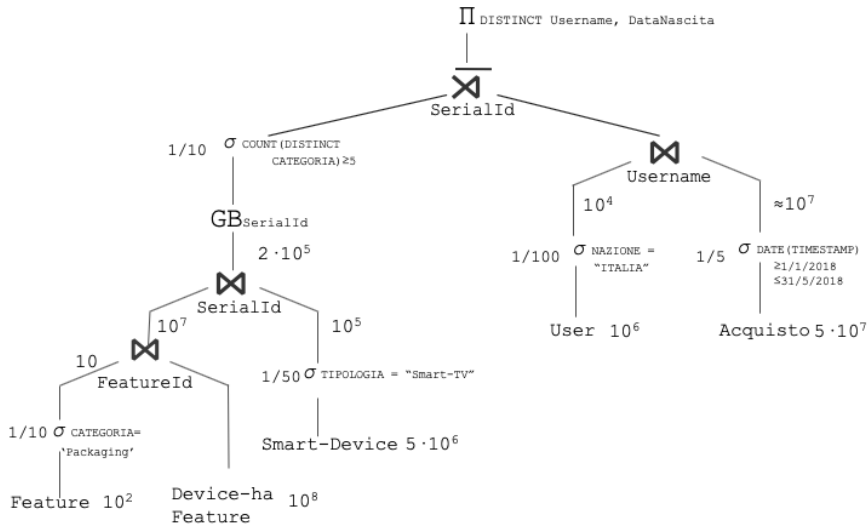
Inoltre, assumere il seguente fattore di riduzione per la condizione di group by:
 HAVING COUNT(DISTINCT F.Categoria) >= 5 $\approx 1/10$

Si consideri la seguente query:

```
select DISTINCT Username, DataNascita
from USER U, ACQUISTO A
where U.Username=A.Username and Nazione='Italia'
and DATE(Timestamp) >= 1/1/2018 and DATE(Timestamp) <= 31/5/2018
and SerialId NOT IN (select SerialId
from SMART-DEVICE D, FEATURE F, DEVICE-HA-FEATURE DHF
where D.SerialId=DHF.SerialId and F.FeatureId=DHF.FeatureId
and F.Categoria='Packaging'
D.Tipologia = 'SmartTV'
group by SerialId
having COUNT(DISTINCT F.Categoria)>=5)
```

La figura sottostante rappresenta il query tree per la query precedente.

Si selezionino le strutture fisiche accessorie per migliorare le prestazioni dell'interrogazione (se possibile).



Scegli una o più alternative:

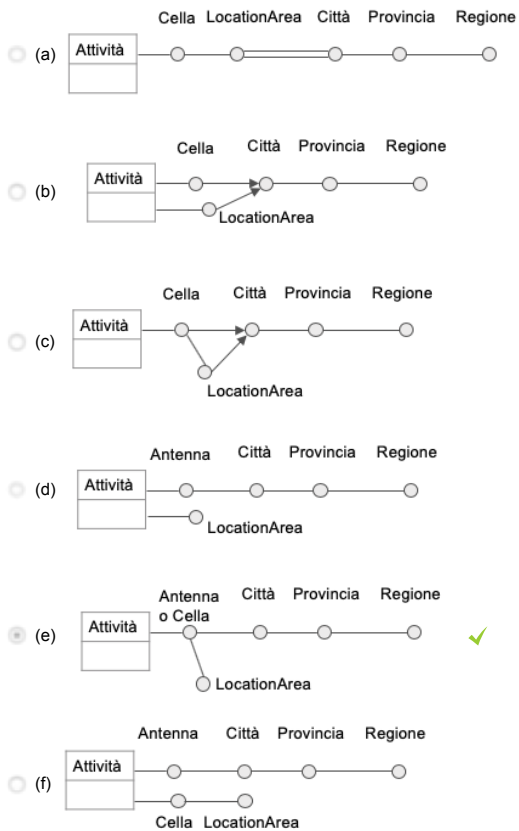
- (a) CREATE INDEX IndexE ON SMART-DEVICE(SerialId) - HASH
- (b) CREATE INDEX IndexD ON SMART-DEVICE(Tipologia) - HASH ✓
- (c) Nessuna - le strutture fisiche accessorie non migliorerebbero le prestazioni dell'interrogazione
- (d) CREATE INDEX IndexC ON DEVICE-HA-FEATURE(FeatureId) - B⁺-Tree
- (e) CREATE INDEX IndexA ON FEATURE(Categoria) - HASH
- (f) CREATE INDEX IndexB ON DEVICE-HA-FEATURE(SerialId) - HASH
- (g) CREATE INDEX IndexF ON USER(Nazione) - HASH ✓
- (h) CREATE INDEX IndexG ON ACQUISTO(Data) - B⁺-Tree

Schema concettuale 1 (1 punto, penalità -15% per risposta sbagliata)

Gli analisti di una società di telefonia mobile sono interessati ad analizzare alcune statistiche sull'utilizzo delle celle telefoniche, quali ad esempio, il numero di telefonate effettuate e la durata media delle telefonate.

- Ogni **antenna** della rete definisce un'area geografica di copertura detta **cella** telefonica.
- La **posizione** geografica delle celle telefoniche può essere trattata in due modi.
- Il primo consiste nel suddividere il territorio in **porzioni** geografiche dette "**location area**".
- Ogni **cella** telefonica appartiene ad una ed una sola "**location area**".
- Il secondo consiste nel considerare la **città** in cui si trova l'**antenna** della cella telefonica, e di conseguenza, la **provincia** in cui si trova la città e la relativa **regione**.
- Una **città** può essere coperta da diverse "**location area**".
- Una "**location area**" può indipendentemente coprire diverse **città**.

Modellare lo schema concettuale che definisce la gerarchia relativa alla posizione di una cella telefonica.

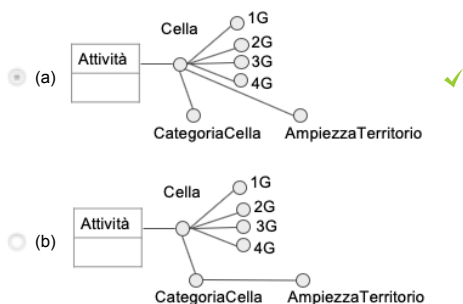


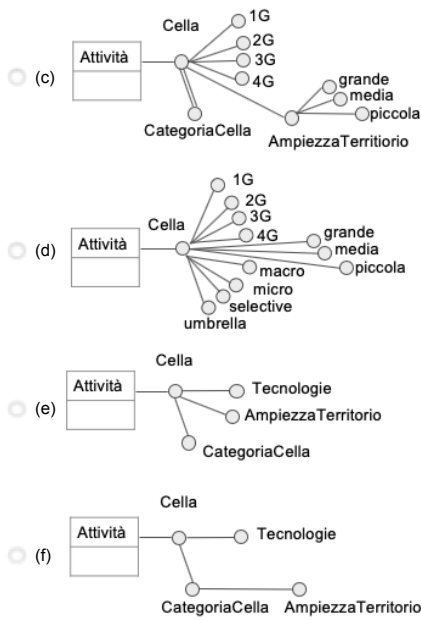
Schema concettuale 2 (1 punto, penalità -15% per risposta sbagliata)

Gli analisti di una società di telefonia mobile sono interessati ad analizzare alcune statistiche sull'utilizzo delle celle telefoniche, quali ad esempio, il numero di telefonate effettuate e la durata media delle telefonate.

- In base alle proprie **caratteristiche**, una **cella** telefonica appartiene ad una delle seguenti **categorie**:
 - "macro-cell",
 - "micro-cell",
 - "selective-cell"
 - "umbrella-cell".
- Per ogni **cella** telefonica sono note le **tecnologie** disponibili (una o più tra le seguenti: 1G, 2G, 3G, 4G).
- É nota infine l'**ampiezza** di territorio coperta dalla cella telefonica
 - grande: oltre 10 km^2
 - media: tra 5 e 10 km^2
 - piccola: meno di 5 km^2

Modellare lo schema concettuale che definisce le caratteristiche di una cella telefonica.



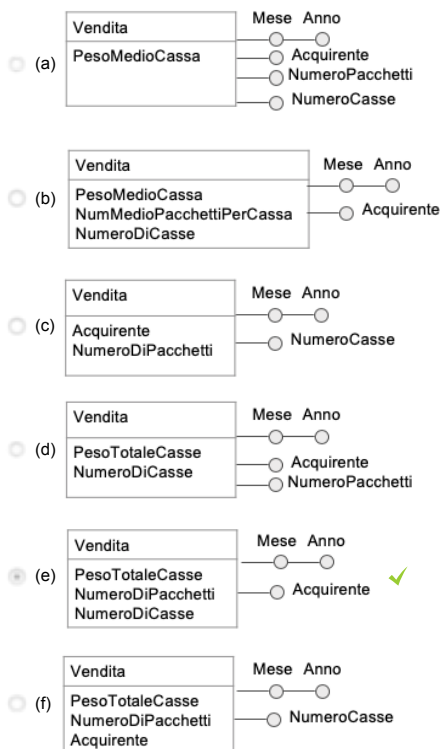


Misure (1 punto, penalità -15% per risposta sbagliata)

Gli analisti di un'industria che vende noccioline all'ingrosso sono interessati ad analizzare statistiche sulle vendite.

- Le noccioline sono vendute in pacchetti, i pacchetti sono raggruppati in casse.
- Ogni cassa può contenere 6 o 12 pacchetti.
- Si vogliono analizzare le statistiche in base al mese di vendita (es. aprile 2020, maggio 2020, ecc.), all'anno di vendita e all'acquirente.
- Le statistiche di interesse sono il numero di casse vendute, il peso medio di una cassa, e il numero medio di pacchetti per cassa.

Progettare lo schema concettuale del data warehouse che soddisfa le specifiche sopra fornite.



Query SQL esteso 1 (4 punti)

Dato il seguente schema relazionale, scrivere le interrogazioni richieste.

Cella (idCella, città, regione)

Data (idData, data, mese, meseDellAnno, semestre, anno)

Attività (idData, OperatoreVirtuale, idCella, numeroDiTransizioni, minutiTelefonate)

Separatamente per ogni città, regione, e anno, calcolare:

- il numero medio giornaliero di transizioni
- la percentuale di transizioni in ogni città rispetto al totale della regione

- all'interno di ogni regione assegnare un rank alle città per numero di transizioni decrescente

```
SELECT città, regione, anno,
       SUM(numeroDiTransizioni)/COUNT(DISTINCT data),
       SUM(numeroDiTransizioni)/SUM(SUM(numeroDiTransizioni)) OVER (PARTITION BY regione,
       RANK() OVER (PARTITION BY regione, anno ORDER BY SUM(numeroDiTransizioni) DESC)
FROM Cella, Data, Attività
WHERE Cella.idCella=Attività.idCella AND Attività.idData=Data.idData
GROUP BY città, regione, anno
```

Query SQL esteso 2 (4 punti)

Dato il seguente schema relazionale, scrivere le interrogazioni richieste.

Cella (idCella, città, regione)

Data (idData, data, mese, meseDellAnno, semestre, anno)

Attività (idData, OperatoreVirtuale, idCella, numeroDiTransizioni, minutiTelefonate)

Separatamente per ogni città, mese e operatore virtuale, calcolare:

- il numero totale di minuti di telefonate
- la percentuale di minuti di telefonate di ogni città rispetto al totale mensile dell'operatore su tutte le città
- il cumulativo mensile di minuti di chiamata dall'inizio dell'anno

```
SELECT città, mese, anno, OperatoreVirtuale,
       SUM(minutiTelefonate),
       SUM(minutiTelefonate)/SUM(SUM(minutiTelefonate)) OVER (PARTITION BY mese, Operatore
       SUM(SUM(minutiTelefonate)) OVER (PARTITION BY città, anno, OperatoreVirtuale
       ORDER BY mese ROWS UNBOUNDED PRECEDING)
FROM Cella, Data, Attività
WHERE Cella.idCella=Attività.idCella AND Attività.idData=Data.idData
GROUP BY città, mese, anno, OperatoreVirtuale
```

Trigger 1 (7 punti)

Sono date le relazioni seguenti (le chiavi primarie sono sottolineate).

SUPERMERCATO(CodSupermercato, Città, NumeroPostiCoda, OraApertura, OraChiusura)

CODA_INGRESSO(CodSupermercato, PosizioneCoda, CodFiscaleCliente)

RICHIESTA_INGRESSO(CodRichiesta, CodSupermercato, CodFiscaleCliente, Data, Ora)

Si vogliono gestire in modo automatico le code in ingresso a una catena di supermercati. Si scriva il trigger per gestire la seguente attività.

Richiesta di inserimento in coda

Un cliente richiede l'inserimento nella coda per uno specifico supermercato mediante una richiesta d'ingresso (inserimento di un record nella tabella RICHIESTA_INGRESSO). Si devono svolgere le seguenti attività.

(a) *Verificare la disponibilità di un posto nella coda.*

L'inserimento in coda avviene assegnando una posizione crescente, a partire da 1 e fino al numero massimo di posti disponibili nella coda, pari al valore dell'attributo NumeroPostiCoda della tabella SUPERMERCATO, diverso per ogni supermercato. Le posizioni libere (se esistono) sono al fondo della coda. La posizione di un cliente nella coda di un supermercato è individuata dall'attributo PosizioneCoda della tabella CODA_INGRESSO. Per verificare la disponibilità di posti, occorre verificare che non sia stato raggiunto il numero massimo di posti disponibili. Se nel supermercato richiesto non è disponibile nessun posto in coda, il trigger termina con un errore.

(b) *Inserimento in coda.*

L'inserimento di un cliente in coda avviene assegnando la posizione immediatamente successiva all'ultimo posto occupato in coda (la posizione nuova è incrementata di 1 rispetto all'ultimo posto occupato). Si consideri anche il caso in cui la coda è vuota.

```

create or replace trigger INSERIMENTO_IN_CODA
after insert on RICHIESTA_INGRESSO
for each row

declare
myPosizione, MaxPosti number;

begin
---trovo l'ultima posizione occupata nella coda del supermercato prescelto
select max (PosizioneCoda) INTO MyPosizione
from CODA_INGRESSO
where CodSupermercato = :new.CodSupermercato;

--- leggo il numero massimo di posti in coda
select NumeroPostiCoda INTO MaxPosti
from SUPERMERCATO
where CodSupermercato = :new.CodSupermercato;

if (myPosizione == MaxPosti) then
--- coda piena; restituisco errore
raise_application_error(...);
end if;

--- processo di inserimento in coda
if (myPosizione is null) then
--- coda vuota; assegno la prima posizione
myPosizione := 1;
else
- -- coda non vuota; assegno la posizione immediatamente successiva all'ultima
myPosizione := myPosizione + 1;
end if;

--- inserisco in coda
insert into CODA_INGRESSO (...)
values (:new.CodSupermercato, myPosizione, :new.CodCliente);
end;

```

Trigger 2 (3 punti)

Sono date le relazioni seguenti (le chiavi primarie sono sottolineate).

SUPERMERCATO(CodSupermercato, Città, NumeroPostiCoda, OraApertura, OraChiusura)
CODA_INGRESSO(CodSupermercato, PosizioneCoda, CodFiscaleCliente)
RICHIESTA_INGRESSO(CodRichiesta, CodSupermercato, CodFiscaleCliente, Data, Ora)

Si vogliono gestire in modo automatico le code in ingresso a una catena di supermercati. Si scriva il trigger per gestire la seguente attività.

Vincolo di integrità sulla durata dell'apertura

Il numero di supermercati di Torino con durata dell'apertura (differenza tra gli attributi OraChiusura e OraApertura) superiore a 18 ore può essere al massimo 10. Ciascuna operazione di modifica della tabella SUPERMERCATO che causi la violazione del vincolo non deve essere eseguita (valutare attentamente tutti gli eventi innescenti sulla tabella SUPERMERCATO).

```

create trigger OrarioAperturaMassimo
after insert or update of OraApertura or update of OraChiusura or update of Citta` on S
declare
X number;

begin
select count(*) into X
from SUPERMERCATO
where Citta` == `Torino'
and (OraChiusura-OraApertura) > 18;

if (X > 10) then
raise_application_error(...)
end if;
end;

```